

**PENGEMBANGAN DAN ANALISIS SISTEM INFORMASI
PENGELOLAAN DATA ANGGOTA
UKM MUSIK SICMA UNY
BERBASIS *FRAMEWORK PHP CODEIGNITER***

SKRIPSI

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Sarjana Pendidikan



**Disusun oleh:
FARIS RIDHA PRIMASTOMO
NIM. 08520241002**

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2014**

HALAMAN PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

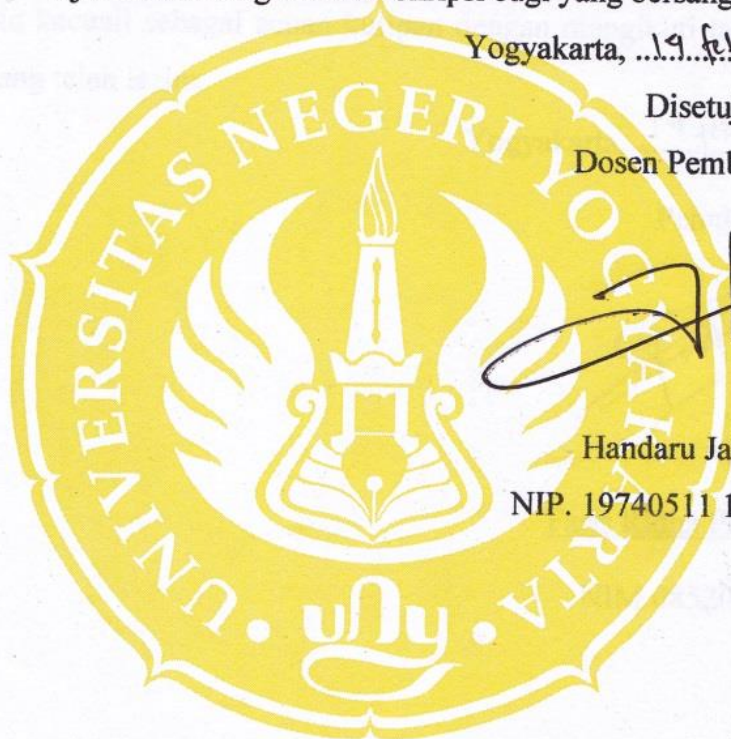
PENGEMBANGAN DAN ANALISIS SISTEM INFORMASI PENGELOLAAN DATA ANGGOTA UKM MUSIK SICMA UNY BERBASIS FRAMEWORK PHP CODEIGNITER DAN MYSQL

Disusun oleh:
Faris Ridha Primastomo
NIM. 08520241002

Telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan
Ujian Akhir Tugas Akhir Skripsi bagi yang bersangkutan.

Yogyakarta, ..19..februari..... 2014

Disetujui,
Dosen Pembimbing,



Handaru Jati, Ph.D
NIP. 19740511 199903 1 002

HALAMAN PERNYATAAN

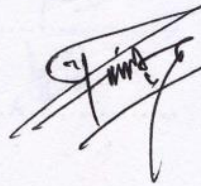
Yang bertanda tangan di bawah ini:

Nama : Faris Ridha Primastomo
NIM : 08520241002
Program Studi : Pendidikan Teknik Informatika
Judul Tugas Akhir Skripsi : Pengembangan dan Analisis Sistem Informasi
Pengelolaan Data Anggota UKM Musik SICMA
UNY Berbasis *PHP Framework Codeigniter* dan
MySql

menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Yogyakarta, 19 Februari 2014

Penulis,



Faris Ridha Primastomo

NIM 08520241002

HALAMAN PENGESAHAN

Tugas Akhir Skripsi

PENGEMBANGAN DAN ANALISIS SISTEM INFORMASI PENGELOLAAN DATA ANGGOTA UKM MUSIK SICMA UNY BERBASIS FRAMEWORK PHP CODEIGNITER DAN MYSQL

Disusun oleh:
Faris Ridha Primastomo
NIM. 08520241002


Telah dipertahankan di depan Tim Penguji Tugas Akhir Skripsi Program Studi
Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta
pada tanggal 3 Maret.....2014

TIM PENGUJI

Nama	Jabatan	Tanda Tangan	Tanggal
Handaru Jati, Ph.D	Ketua Penguji		<u>1/4/2014</u>
Dr. Ratna Wardani	Sekretaris		<u>1/4/2014</u>
Nurkhamid, Ph.D	Penguji		<u>13/3/2014</u>

Yogyakarta, 7 April..... 2014
Fakultas Teknik Universitas Negeri Yogyakarta
Dekan,




Dr. Moch. Bruri Triyono
NIP. 19560216 198603 1 003

MOTTO

Segala sesuatu yang diawali dengan ketidak jujuran pada akhirnya akan hancur

(Romo)

Manusia bisa menjadi apa saja tergantung dari seberapa besar usaha (usaha dan doa) dan waktu yang dikorbankan untuk mendapatkannya

Hidup adalah pilihan, pilihlah dan bertanggung jawablah dengan pilihanmu

(Bundo)

PERSEMBAHAN

Persembahan untuk

Romo dan Bundo

ABSTRAK

Penelitian ini bertujuan menghasilkan sekaligus menganalisis kualitas dari Sistem Informasi Pengelolaan Data Anggota UKM MUSIK SICMA UNY Berbasis *Framework* PHP *Codeigniter* dan Mysql agar diketahui kelayakannya untuk digunakan pengguna akhir. Aspek kualitas perangkat lunak yang diteliti berdasarkan kaidah *Software Quality*. Hasil dari penelitian ini adalah sistem informasi pengelolaan data yang kualitasnya teruji.

Prosedur penelitian yang dilakukan peneliti sesuai dengan kaidah *software engineering* meliputi analisis kebutuhan, tahap desain, implementasi dan analisis faktor kualitas perangkat lunak berdasarkan kaidah *software quality* khususnya faktor *correctness*, *usability*, *integrity*, *efficiency*, *maintainability*, *reliability*, *interoperability* dan *portability*. Analisis faktor *correctness* dilakukan dengan metode KLOC, analisis *usability* menggunakan kuisioner *System Usability Scale* (SUS) dengan responden anggota UKM Musik SICMA UNY, analisis *integrity* menggunakan *Web Vulnerability Scanner*, analisis *efficiency* menggunakan *YSloow* dan *webpagetest*, analisis *maintainability* menggunakan ukuran metrik, analisis *reliability* menggunakan tool berupa *software* dan pembuktian secara *load tester*, analisis *interoperability* menggunakan *cohesion metric* (LCC) dan *coupling metric* (CBO, RFC, NOC) dan analisis *portability* diuji menggunakan pembuktian secara *load tester*.

Analisis aspek *correctness*, *usability*, *integrity*, *efficiency*, *maintainability*, *reliability*, *interoperability* dan *portability* menghasilkan nilai standar kualitas yang dianjurkan. Analisis *correctness* menghasilkan nilai standar kualitas yang diharuskan pada rentang 0-40 error/ KLOC, analisis *usability* menghasilkan nilai 2442.5 dengan kategori baik, analisis *integrity* membuktikan bahwa sistem lolos uji dari celah *SQL Injection*, *Blind_SQL Injection* dan *XSS (Cross Site Scripting)*, analisis *efficiency* menunjukkan sistem sudah memenuhi standar performa dengan mendapatkan grade A pada *Yslow* dan nilai 100/100 pada *webpagetest*, analisis *maintainability* menunjukkan sistem memenuhi pada aspek *instrumentation*, *consistency* dan *simplicity*, analisis *reliability* menunjukkan sistem lolos pada uji *correct link processing*, *user input validation and recovery*, dan *error recovery*, analisis *interoperability* sudah memenuhi standar *cohesion metric* dan *coupling metric* dan analisis *portability* menunjukkan bahwa sistem dapat bekerja dengan baik tanpa error pada 4 *web browser*. Hasil analisis pada semua aspek menunjukkan bahwa aplikasi yang dikembangkan lolos dari uji kualitas perangkat lunak.

Kata kunci: Sistem informasi, *software quality*, *analisis software*.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi dengan judul “Pengembangan dan Analisis Sistem Informasi Pengelolaan Data Anggota UKM Musik Sicma UNY Berbasis *Framework PHP Codeigniter*” sebagai salah satu syarat untuk memperoleh gelar Sarjana Pendidikan di Fakultas Teknik Universitas Negeri Yogyakarta. Penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dan peran dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Handaru Jati, Ph. D. selaku Dosen Pembimbing Tugas Akhir Skripsi yang telah memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini;
2. Dr. Ratna Wardhani dan Nurkhamid, Ph.D selaku Sekretaris dan Penguji yang telah memberikan koreksi perbaikan secara komprehensif terhadap Tugas Akhir Skripsi ini;
3. Dr. Ratna Wardhani selaku Ketua Program Studi Pendidikan Teknik Informatika beserta Dosen dan staf yang telah memberikan bantuan dan fasilitas selama proses penyusunan Tugas Akhir Skripsi ini;
4. Dra. Umi Rochayati, M.T. selaku Dosen Pembimbing Akademik yang selalu memberikan pengarah selama perkuliahan;
5. Dr. Moch. Bruri Triyono, selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang telah memberikan persetujuan pelaksanaan Tugas Akhir Skripsi ini;
6. Ayah saya Sujarno dan Ibu saya Siti Amini tercinta yang selalu memberikan doa, semangat dan bantuan berupa moral maupun material meskipun sering kali berbeda pandangan;
7. Saudara saya Rizkiaji Primastomo, Chusnul Liyna Primastomo dan Ikhsan Hanaan Primastomo kalian joss!!;
8. Teman-teman UKM Musik Sicma UNY yang telah membantu dalam penelitian;

9. Teman-teman SWKY yang selalu memberikan semangat berupa sentilan pedas dan dukungan moral;
10. Teman-teman KOMBRE atas dukungan dan bantuan yang diberikan, saudara Anggit beserta Keluarga Kulon Progo terimakasih atas tumpangannya;
11. Kos Pakar on7 dan warga pandega karya kalian joss!!;
12. Semua pihak yang tidak dapat penyusun sebutkan satu persatu yang telah membantu penyusun hingga tersusunnya laporan Tugas Akhir Skripsi ini.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini, namun penulis tetap berharap semoga laporan ini berguna dan bermanfaat baik bagi penulis maupun pembaca.

Yogyakarta, 1 April 2014

Penulis,

Faris Ridha Primastomo

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PERNYATAAN.....	iii
HALAMAN PENGESAHAN.....	iv
MOTTO	v
PERSEMBAHAN.....	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN.....	1
A. Latar Belakang	1
B. Identifikasi Masalah.....	3
C. Batasan Masalah	4
D. Rumusan Masalah.....	5
E. Tujuan Penelitian	5
F. Manfaat Penelitian	6
1. Bagi Pengguna.....	6
2. Bagi Peneliti	6
BAB II KAJIAN TEORI	7
A. Deskripsi Teori	7
1. <i>Framework PHP Codeigniter</i>	7
2. Sistem Informasi.....	8
3. <i>Software Engineering/</i> Rekayasa Perangkat Lunak	8
4. <i>Software Quality/</i> Kualitas Perangkat Lunak	9
B. Penelitian yang relevan	22
BAB III METODE PENELITIAN	23
A. Desain Penelitian	23

1.	Metode Penelitian	23
2.	Tempat dan Waktu Penelitian	23
B.	Alur Penelitian	24
1.	Analisis Kebutuhan	24
2.	Desain	25
3.	Tahap Implementasi	25
4.	Tahap pengujian	26
5.	Instrumen Penelitian	28
BAB IV	HASIL DAN PEMBAHASAN	30
A.	Tahap Analisis Kebutuhan	30
1.	Analisis Proses	30
2.	Analisis Kebutuhan <i>Hardware</i> dan <i>Software</i>	30
B.	Tahap Desain	31
1.	Perancangan Unified Modelling Language (UML)	31
2.	Perancangan user interface (Antar Muka Pengguna)	34
3.	Perancangan Basis Data	35
C.	Tahap Implementasi	36
1.	Implementasi <i>User Interface</i> (Antar Muka Pengguna)	36
2.	Implementasi Basis Data	43
D.	Tahap Pengujian	46
1.	Hasil pengujian Correctness	46
2.	Hasil Pengujian Usability	48
3.	Hasil pengujian Integrity	50
4.	Hasil pengujian Maintainability	53
5.	Hasil pengujian Efficiency	56
6.	Hasil pengujian Reliability	57
7.	Hasil pengujian Interoperability	60
8.	Hasil Pengujian <i>Portability</i>	65
BAB V	KESIMPULAN DAN SARAN	67
A.	Kesimpulan	67
B.	Saran	68

DAFTAR PUSTAKA	69
LAMPIRAN.....	72
A. <i>System Usability Scale (SUS)</i>	73
B. Perancangan Tampilan.....	75
C. Impelentasi Tampilan	76
D. Perancangan Database	77

DAFTAR TABEL

Table 1. Cara Penghitungan KLOC menurut Ahli	11
Table 2. Perkiraan Jumlah Error McConnell.....	11
Table 3. Black-box Testing Integrity.....	13
Table 4. Instrumen Maintainability	14
Table 5. Instrumen <i>System Usability Scale</i> (SUS)	16
Table 6. <i>Standar System Usability Scale</i>	17
Table 7. Instrumen <i>Portability</i>	22
Table 8. Hasil Penghitugan KLOC.....	48
Table 9. Penghitungan Kuisisioner <i>System Usability Scale</i> (SUS).....	49
Table 10. Nilai Standar <i>System Usability Scale</i> (SUS)	50
Table 11. Hasil Pengujian Integrity (Web Vulnerability Scanner)	51
Table 12. Hasil Angket Pendukung Integrity Testing	52
Table 13. Hasil Snalisis Aspek Maintainability	55
Table 14. Hasil Penghitungan LCC.....	61
Table 15. Hasil Pengukuran CBO	62
Table 16. Hasil Pengukuran NOC	63
Table 17. Hasil Pengukuran RFC.....	64
Table 18. Hasil Pengujian Metrik Interoperability	65
Table 19. Hasil Pengujian Portability.....	66

DAFTAR GAMBAR

Gambar 1. Model Kualitas Perangkat Lunak menurut McCall	10
Gambar 2. Waterfall	24
Gambar 3. <i>Use Case User</i>	31
Gambar 4. <i>Use Case Admin</i>	32
Gambar 5. <i>Activity Diagram User</i>	33
Gambar 6. <i>Activity Diagram Admin</i>	34
Gambar 7. Perancangan <i>User Interface</i> Form Pengisian Data.....	35
Gambar 8. Rancangan Desain Basis Data	35
Gambar 9. Form Login User	36
Gambar 10. Form Registrasi Mandiri.....	37
Gambar 11. Form Lihat Profil	38
Gambar 12. Form Edit Profil.....	38
Gambar 13. Form Cari-Lihat Data Anggota.....	39
Gambar 14. Form Detail Data Anggota.....	39
Gambar 15. Form Login Admin	40
Gambar 16. Form Lihat Admin	40
Gambar 17. Form Update Admin	41
Gambar 18. Form Tambah Admin	41
Gambar 19. Form Lihat dan Cari Data (Admin)	42
Gambar 20. Form Tambah Data (<i>Admin</i>).....	42
Gambar 21. Implementasi Tabel Admin	43
Gambar 22. Implementasi Tabel Data.....	43

Gambar 23. Implementasi Tabel Fakultas	44
Gambar 24. Implementasi Tabel User	44
Gambar 25. Implementasi Tabel Organisasi	44
Gambar 26. Implementasi Tabel Player	45
Gambar 27. Implementasi Tabel Eo	45
Gambar 28. Implementasi Tabel Agama	45
Gambar 29. Implementasi Tabel Ortu	46
Gambar 30. Hasil Pengujian KLOC dengan SLOC	47
Gambar 31. Hasil Pengujian <i>Web Vulnerability Scanner</i>	51
Gambar 32. Peringatan Gagal Login (<i>username</i> dan <i>password</i> salah)	53
Gambar 33. Peringatan Gagal Login	53
Gambar 34. Peringatan saat Registrasi Mandiri	54
Gambar 35. Hasil Uji Halaman data (admin mode)	56
Gambar 36. Hasil Grade YSlow	56
Gambar 37. Pengujian Performance Web dengan Webpagetest	57
Gambar 38. Hasil pengujian Xenu Link Sleuth 1.3.8.....	58
Gambar 39. Hasil Uji Validasi Login User	59
Gambar 40. Hasil Pengujian Login User (2)	59
Gambar 41. Hasil Pengujian Form Nim (Input Data)	60
Gambar 42. Gambar Pengukuran LCC.....	61
Gambar 43. Pengukuran CBO	62
Gambar 44. Pengukuran NOC.....	63
Gambar 45. Gambar Fungsi RFC	64

DAFTAR LAMPIRAN

Lampiran 1. Contoh Kuisisioner <i>System Usability Scale</i> (1).....	73
Lampiran 2. Contoh Kuisisioner <i>System Usability Scale</i> (2).....	74
Lampiran 3. Perancangan Form Login.....	75
Lampiran 4. Perancangan Form Isi Konten.....	75
Lampiran 5. Gambar Tampilan Login User	76
Lampiran 6. Gambar Tampilan Profil	76
Lampiran 7. Gambar Tampilan <i>view_data_all</i>	77
Lampiran 8. Gambar Daftar Tabel Database.....	77
Lampiran 9. Gambar Tabel Database "Admin"	78
Lampiran 10. Daftar Tabel Database "User"	78
Lampiran 11. Daftar Tabel Database "Data"	78
Lampiran 12. Daftar Tabel Database "Fakultas"	79
Lampiran 13. Daftar Database Tabel "Prodi"	79
Lampiran 14. Daftar Tabel Database "Organisasi"	79
Lampiran 15. Daftar Tabel Database "Player"	79
Lampiran 16. Daftar Tabel Database "Eo"	80

BAB I

PENDAHULUAN

A. Latar Belakang

UKM merupakan unit kegiatan mahasiswa yang berdiri di bawah naungan universitas baik dalam lingkup fakultas maupun lingkup universitas secara menyeluruh. Berdasarkan laporan pengurus UKM Musik Sicma UNY pada sidang akhir tahun, *musyang* (Musyawarah Anggota, 2010), menyatakan bahwa kesulitan administrasi yang belum mendapatkan solusi dari awal mula berdirinya UKM adalah pendataan anggota. Hal ini terjadi dikarenakan dalam AD/ ART UKM Musik Sicma UNY menyatakan bahwa anggota yang telah lulus dari UNY tetap menjadi anggota UKM Sicma dan berubah keanggotaan menjadi alumni. Dan dalam kenyataannya pendataan dan pembaharuan data anggota yang dilakukan sampai saat ini mengharuskan anggota datang kesekretariatan UKM Musik Sicma UNY, hal ini dirasa tidak efisien karena membutuhkan waktu dan tenaga yang lebih besar dalam melakukan pendataan dan pembaharuan data serta dengan adanya tambahan data setiap tahun dari anggota baru.

Pendataan anggota yang dilakukan hingga saat ini masih menggunakan cara konvensional menggunakan kertas, hal ini akan menimbulkan kesulitan jika data yang ada terus bertambah setiap tahun, sedangkan sifat kertas yang rapuh dan rentan rusak dapat mengakibatkan rusaknya data jika kertas tidak tersip dengan benar.

Penyimpanan kertas yang terus menerus bertambah setiap tahun lambat laun akan memakan tempat yang setiap tahun akan bertambah. Meskipun terdapat

tempat penyimpanan yang layak akan tetapi seiring berjalannya waktu tumpukan kertas akan menjadi semakin banyak dan terjadi pemborosan tempat penyimpanan.

Selain itu tumpukan kertas yang banyak menyebabkan kesulitan dalam pengelolaan data seperti pencarian data anggota dan perubahan data yang sulit dilakukan karena harus mencari di antara tumpukan kertas. Hal ini akan membuat pengelolaan data tidak efisien dan membutuhkan waktu dan tenaga yang semakin besar.

Pendataan yang dilakukan secara konvensional juga memiliki kelemahan pada duplikasi data. Duplikasi data dapat terjadi karena pengisian data dilakukan secara konvensional dan faktor manusia yang cenderung membuat kesalahan dengan memasukkan data yang sama. Meskipun terdapat *unique* data berupa *nim* maupun nomor Kartu Tanda Anggota (KTA) akan tetapi faktor manusia tidak dapat diandalkan dalam pengisian data secara konsisten.

Pada UKM Musik Sicma UNY secara umum dapat dibagi menjadi 2 bagian yakni *player* dan *eo*, dan pada *player* dibagi lagi menjadi beberapa *player* sesuai jumlah *skill* (*guitaris, bassis, vokal, drummer, keyboardis, dll*). Sehubungan dengan pendataan anggota UKM Musik Sicma UNY pada setiap tahun dibentuk *band* dan *event organizer* dengan jumlah sesuai dengan jumlah anggota. Hal ini akan sulit dilakukan jika pendataan dilakukan secara konvensional dan pengelolaan data dalam bentuk statistik sesuai dengan pembagian *skill* sulit untuk diketahui.

Permasalahann selanjutnya dalam pendataan yang dilakukan secara konvensional dengan kertas adalah kesulitan dalam pengarsipan data dalam bentuk cetak seperti terjadi pada akhir pengurusan setiap tahun pada lembar LPJ (Lembar Pertanggung Jawaban). Dalam hal ini pengarsipan yang dilakukan secara konvensional lebih membutuhkan tenaga dan waktu dengan pekerjaan yang tidak efisien.

Dari penjelasan di atas maka dibutuhkan sebuah sistem informasi *online* yang dapat diakses di mana saja oleh anggota UKM. Sistem informasi *online* dibutuhkan supaya dapat diakses di mana saja terutama bagi anggota alumni yang telah meninggalkan UKM Sicma UNY. Sistem informasi yang dibutuhkan dapat memberikan pelayanan pengelolaan data anggota UKM dengan kemampuan dapat melakukan penambahan data, perubahan data, pencarian data, penghapusan data, pengelolaan data dalam bentuk statistik dan pengelolaan data dalam media cetak.

Sistem informasi *online* yang digunakan haruslah dapat diandalkan dan terdapat sistem keamanan yang memadai dikarenakan data yang ada merupakan data vital dari UKM, sehingga dengan adanya kebutuhan keamanan data dan kemudahan *user* dalam melakukan pengelolaan data, maka sistem informasi ini harus diuji sesuai dengan standar kelayakan fungsi *software*. Dalam hal ini pengujian akan dilakukan dari aspek *Correctness, Maintainability, Integrity, Usability, Efficiency, Reliability, Interoperability* dan *Portability*.

B. Identifikasi Masalah

Berdasarkan uraian di atas tentang latar belakang masalah yang ada, maka dapat dilakukan identifikasi masalah, yaitu:

1. pendataan anggota yang harus dilakukan di sekretariat UKM tidak efisien;
2. kesulitan dalam melakukan pengelolaan data anggota UKM yang terus bertambah setiap tahun karena pendataan masih dilakukan secara konvensional dengan kertas;
3. umur kertas yang tidak tahan lama dan rapuh sehingga berpotensi terjadi kerusakan data;
4. terjadi pemborosan tempat penyimpanan kertas dikarenakan data bertambah setiap tahun;
5. terjadinya duplikasi data dalam pendataan anggota UKM Musik UNY;
6. pemborosan tempat penyimpanan yang semakin bertambah setiap tahun;
7. kesulitan menampilkan data dalam bentuk statistik;
8. belum adanya sistem informasi untuk membantu pengelolaan data anggota UKM yang dapat diakses secara *online*.

C. Batasan Masalah

Dari identifikasi masalah di atas penulis membatasi masalah yang akan diselesaikan dalam penelitian ini sebagai berikut:

1. mengembangkan aplikasi untuk pengelolaan data anggota UKM sehingga dapat mengakomodasi pendataan anggota yang jumlahnya bertambah setiap tahun;
2. terjadinya duplikasi data dalam pendataan anggota UKM Musik UNY;
3. kesulitan menampilkan data dalam bentuk statistik;
4. dibutuhkan sebuah sistem informasi *online* yang dapat membantu pelayanan pengelolaan data;

5. dibutuhkan pengujian sistem informasi yang dikembangkan sesuai dengan standar kelayakan *software*, dan pengujian dilakukan dari aspek *Correctness*, *Maintainability*, *Integrity*, *Usability*, *Efficiency*, *Reliability*, *Interoperability* dan *Portability*.

Pada penelitian ini difokuskan pada perancangan dan analisis sistem informasi pengelolaan data anggota UKM khususnya UKM Band Sicma UNY. Aplikasi ini memiliki fungsi utama untuk melakukan penambahan, perubahan, penghapusan, pencarian data serta menampilkan statistik.

D. Rumusan Masalah

Berdasarkan uraian di atas tentang pentingnya pengelolaan data yang baik dan efisien, permasalahan dititikberatkan pada komputerasi sistem pengelolaan data Anggota UKM Band Sicma UNY. Permasalahan yang diselesaikan dalam penelitian ini dapat dirumuskan sebagai berikut:

1. bagaimana membuat perangkat lunak Sistem Informasi Pengelolaan Data Anggota UKM Band Sicma UNY?
2. bagaimana tingkat kualitas perangkat lunak yang dikembangkan dari sisi : *Correctness*, *Maintainability*, *Integrity*, *Usability*, *Efficiency*, *Reusability*, *Interoperability* dan *Portability*?

E. Tujuan Penelitian

Tujuan penelitian analisis pengembangan perangkat lunak sistem informasi pengelolaan data anggota UKM Band Sicma ini adalah:

1. Untuk mengembangkan perangkat lunak sistem informasi untuk pengelolaan data anggota UKM.

2. Untuk menguji dan mengetahui kualitas perangkat lunak yang dikembangkan dari sisi *Correctness*, *Maintainability*, *Integrity*, *Usability*, *Efficiency*, *Reusability*, *Interoperability* dan *Portability*.

F. Manfaat Penelitian

Pengembangan perangkat lunak sistem informasi pengelolaan data anggota ini diharapkan dapat bermanfaat bagi beberapa pihak antara lain:

1. Bagi Pengguna

- a) Mengenalkan sistem pendataan anggota UKM yang baik.
- b) Mempermudah UKM untuk melakukan pendataan anggota (*study tracer*).
- c) Mempermudah dalam mendapatkan informasi sesama anggota UKM.

2. Bagi Peneliti

- a) Mengenal dan memahami lebih jauh teknologi pengembangan perangkat lunak.
- b) Mengetahui teknik mengembangkan perangkat lunak sistem informasi pengelolaan data anggota UKM.
- c) Mengetahui teknik pengujian kualitas perangkat lunak.

BAB II

KAJIAN TEORI

A. Deskripsi Teori

1. *Framework PHP Codeigniter*

Codeigniter adalah sebuah *framework* untuk *web* yang dibuat dalam format PHP. Format yang dibuat ini selanjutnya dapat digunakan untuk membuat sistem aplikasi *web* yang kompleks. *Framework* secara umum adalah sebuah susunan atau rangkaian kerja yang tetap dan dibuat sedemikian rupa yang kemudian dapat digunakan kembali dalam sebuah aktivitas kerja yang lain tapi tetap dalam satu area kerja dengan rangkaian kerja yang sebelumnya.

Codeigniter menggunakan model M-V-C *framework*, mode M-V-C adalah singkatan untuk “Model-View-Controller”. Model Framework ini dapat mempermudah pembedaan antara tampilan dan program yang secara khusus akan sangat baik untuk suatu proyek di mana *designer* bekerja dengan file template yang akan memperkecil banyaknya kode pada setiap template karena sudah dipisahkan dengan baik. Keuntungan penggunaan *codeigniter* dalam pengembangan *web* di antaranya (Wiswakarma, 2010, hal. 2-5).

- a) *Opensource (free)*
- b) Sangat ringan dijalankan di semua platform
- c) *Codeigniter* menggunakan M-V-C (*Model-View-Controler*)
- d) *Codeigniter* menciptakan URL yang *friendly*
- e) Dikemas dalam sebuah *framework* yang lengkap
- f) Fungsi dalam *codeigniter* dapat ditambahkan

- g) Mempunyai dokumentasi (*user guide*)
- h) Mempunyai komunitas pengguna yang besar.

Dengan adanya keunggulan yang dimiliki sesuai uraian di atas maka penulis menggunakan *codeigniter* sebagai *framework* yang digunakan dalam pengembangan sistem informasi pengelolaan data anggota UKM Musik Sisma UNY.

2. Sistem Informasi

Menurut Sutabri sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasional organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan pihak luar (Sutabri, 2012, hal. 38).

Sedangkan pengertian Sistem Informasi menurut Budi Sutedjo adalah kumpulan elemen yang saling berhubungan satu sama lain untuk membentuk suatu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi tersebut (Darmastuti, 2013). Sistem informasi juga dapat diartikan perangkat lunak berbasis *web*. Halaman-halaman *web* diambil oleh *browser* adalah perangkat lunak yang menggabungkan instruksi dieksekusi (misalnya, CGI, HTML, Perl, atau Java), dan data (S.Pressman, 2001, hal. 38)

3. *Software Engineering*/ Rekayasa Perangkat Lunak

Menurut Fritz Baeur dalam Pressman (2001, hal. 20), *software engineering* adalah pembentukan dan penggunaan prinsip-prinsip rekayasa untuk memperoleh

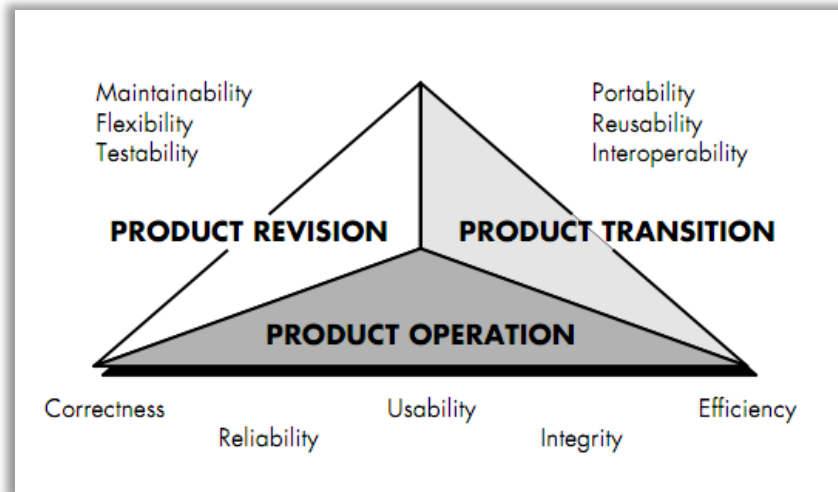
perangkat lunak secara ekonomis yang handal dan bekerja secara efisien pada mesin nyata.

Demikian juga IEE juga memberikan definisi tentang rekayasa perangkat lunak dengan lebih komprehensif yang menyatakan Rekayasa Perangkat Lunak : (1) Penerapan sistematis, disiplin, pendekatan kuantitatif untuk pengembangan, operasi, dan pemeliharaan perangkat lunak, yaitu, penerapan rekayasa perangkat lunak. (2) Studi pendekatan yang lebih dalam tentang rekayasa perangkat lunak (S.Pressman, 2001, hal. 48).

4. *Software Quality*/ Kualitas Perangkat Lunak

Software Quality/ Kualitas Perangkat Lunak memiliki banyak definisi kualitas perangkat lunak telah diusulkan dalam literatur. Dalam bukunya Pressman menjelaskan kualitas perangkat lunak didefinisikan sebagai Kesesuaian secara eksplisit menyatakan persyaratan fungsional dan kinerja, standar pengembangan secara eksplisit didokumentasikan, dan karakteristik implisit yang diharapkan dari semua perangkat lunak yang dikembangkan secara profesional (S.Pressman, 2001, hal. 228).

Sehubungan dengan kualitas perangkat lunak, ada beberapa teori ahli yang menjelaskan faktor kualitas perangkat lunak (*Quality Model*) sebagai berikut : Menurut McCall dan rekan-rekannya (S.Pressman, 2001, hal. 537) terdapat 3 aspek penting dari suatu produk *software*, yaitu : karakteristik operasional, kemampuan perubahan ketika *software* sudah berjalan, dan kemampuan beradaptasi terhadap lingkungan baru seperti dijelaskan pada gambar 1.



Gambar 1. Model Kualitas Perangkat Lunak menurut McCall
(Kualitas Perangkat Lunak McCall (Roger S.Pressman, 2001)).

Berdasarkan uraian di atas, McCall dkk dalam Pressman (2001, hal. 537) menyediakan beberapa deskripsi *software quality factor* diantaranya: *Correctness*, *Reliability*, *Efficiency*, *Integrity*, *Usability*, *Maintainability*, *Flexibility*, *Testability*, *Portability*, *Reusability*, *Interoperability*, sedangkan Pengukuran kualitas perangkat lunak menurut Tom Gilb terdapat 4 deskripsi : *Correctness*, *Maintainability*, *Integrity*, *Usability* (S.Pressman, 2001, hal. 124). Dan pada penelitian ini pengujian yang dilakukan menggunakan menggunakan 8 deskripsi kualitas perangkat lunak, diantaranya : *Correctness*, *Maintainability*, *Integrity*, *Usability*, *Efficiency*, *Reliability*, *Interoperability* dan *Portability*.

a) Aspek *Correctness*

Kebenaran adalah persyaratan minimum dari sebuah perangkat lunak dan merupakan tujuan penting dari pengujian. Menurut Tom Gilb *correctness*

didefinisikan dengan program harus bekerja dengan benar dan merupakan tingkat dimana perangkat lunak yang bekerja sesuai dengan fungsi yang dibutuhkan (Roger S.Pressman, 2001, hal. 124). Sedangkan menurut McCall aspek kebenaran adalah sejauh mana program memenuhi spesifikasi dan memenuhi tujuan misi konsumen/ *user* (Pressman, 2010, hal. 402). Faktor kualitas *correctness* secara umum dapat diukur dengan analisis *defect* per KLOC (cacat / *error* pada setiap KLOC/Kilo Line of Code). Ada beberapa teori ahli untuk menghitung jumlah KLOC (Roger S.Pressman, 2001).

Table 1. Cara Penghitungan KLOC menurut Ahli

No	Metode	Rumus
1	Watson – Felix Model	$E = 5.2 \times (KLOC)^{0.91}$
2	Bailey-Basili Model	$E = 5.5 + 0.73 \times (KLOC)^{1.16}$
3	Boehm Simple Model	$E = 3.2 \times (KLOC)^{1.05}$
4	Doty Model(untuk KLOC>9)	$E = 5.28 \times (KLOC)^{1.047}$

Setelah mendapatkan hasil pengujian KLOC kemudian di bandingkan dengan teori pembandingan untuk batas *error* yang diperbolehkan dalam program dengan total jumlah *KLOC*, seperti yang dikemukakan McConnell (2004, hal. 698) tentang batas toleransi *error* program berdasarkan jumlah *Line of Code/ LOC* Program.

Table 2. Perkiraan Jumlah Error McConnell

Ukuran Project (Line of Code / LOC)	Perkiraan Jumlah <i>Error</i>
Lebih kecil dari 2K	0 – 25 <i>error</i> / <i>KLOC</i>

Ukuran Project (Line of Code / LOC)	Perkiraan Jumlah <i>Error</i>
2K – 16K	0 – 40 <i>error / KLOC</i>
16K – 64K	0.5 – 50 <i>error / KLOC</i>
64K – 512K	2 – 70 <i>error / KLOC</i>
Lebih dari 512K	4 – 100 <i>error / KLOC</i>

b) Aspek *Integrity*

Pengujian pada aspek *Integrity* berkaitan dengan keamanan sistem perangkat lunak, yaitu persyaratan untuk mencegah akses pada *user* yang tidak berkepentingan, untuk membedakan antara mayoritas *user* yang dapat melihat informasi (“*read only* ”) , dan *user* yang diizinkan untuk menambah dan mengubah data (*read-write*). Sedangkan menurut McCall *Integrity* didefinisikan dengan sejauh mana akses ke perangkat lunak atau data oleh orang yang tidak berwenang dapat dikontrol (Pressman, 2010, hal. 402). Demikian juga Tom Gilb yang menjelaskan bahwa Atribut *integrity* terhadap keamanan dengan adanya serangan baik disengaja maupun tidak (S.Pressman, 2001, hal. 125).

Pengujian pada aspek *integrity* dapat diuji menggunakan *black-box testing* yang menguji tentang batasan hak akses *user*, dan menggunakan *Acunetix Web Vulnerability Scanner*. Kuisisioner *black-box testing* dapat dilihat pada Tabel 3.

Table 3. Black-box Testing Integrity

No.	Kuisisioner
1	Apakah terdapat pembagian <i>user</i> dan hak akses dalam aplikasi ini ?
2	“ <i>super-admin</i> ” dapat menambah “ <i>admin</i> ” dan “ <i>user</i> ” baru ?
3	“ <i>super-admin</i> ” dapat mengakses halaman “ <i>admin</i> ” dan “ <i>user</i> ” ?
4	“ <i>super-admin</i> ” dapat melakukan perubahan data pada “ <i>admin</i> ” dan “ <i>user</i> ” ?
5	“ <i>super-admin</i> ” dapat melakukan pencarian data <i>user</i> (biasa) ?
6	“ <i>super-admin</i> ” dapat melakukan perubahan data “ <i>admin</i> ” dan “ <i>user</i> ” ?
7	“ <i>admin</i> ” dapat mengakses halaman <i>admin</i> ?
8	“ <i>admin</i> ” dapat melakukan pencarian data <i>user</i> (biasa)?
9	“ <i>admin</i> ” dapat melakukan perubahan data <i>user</i> (biasa)?
10	“ <i>user</i> ” (biasa) tidak dapat mengakses halaman “ <i>admin</i> ”.
11	“ <i>user</i> ” (biasa) tidak dapat melakukan perubahan data “ <i>admin</i> ”
12	“ <i>user</i> ” (biasa) dapat mengubah data pribadi.
13	“ <i>user</i> ” (biasa) tidak dapat melakukan perubahan data pada <i>user</i> (biasa) lain.
14	“ <i>user</i> ” (biasa) dapat melakukan pencarian (<i>tracking</i>) dan melihat informasi <i>user</i> (biasa) lain.
15	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat mengakses halaman <i>web</i> kecuali halaman pendaftaran.
16	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat melihat data.
17	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat melakukan pencarian data.
18	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat melakukan perubahan data

c) Aspek Maintainability

Aspek Perawatan (*Maintainability*) berhubungan dengan kemudahan perbaikan jika ada kesalahan, penyesuaian terhadap perubahan lingkungan atau peningkatan sesuai permintaan pemakai (Paasonen, 2011). Aspek *maintanability*

dijelaskan sebagai usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program (McCall, 1977). Sedangkan syarat ISO 9126 mendefinisikan *maintainability* sebagai kemudahan sebuah perangkat lunak untuk dipahami, dikembangkan, dan diperbaiki. Beberapa indikator kriteria yang dinilai antara lain adalah *consistency*, *simplicity*, *conciseness*, *self-descriptiveness*, dan *modularity*.

Pengujian *maintainability* menggunakan ukuran-ukuran (*metrics*) yang kemudian pengujian dilakukan uji secara operasional (Land). Berikut instrumen yang digunakan untuk pengujian aspek *maintainability*. Metric *maintainability* dapat dilihat pada Tabel 4.

Table 4. Instrumen Maintainability

Aspek	Aspek yang dinilai	Hasil yang akan diperoleh
<i>Instrumentation</i>	Terdapat peringatan pada sistem pengolah data untuk mengidentifikasi kesalahan	Ketika ada kesalahan yang dilakukan oleh user, maka sistem akan mengeluarkan peringatan untuk mengidentifikasi kesalahan.
<i>Consistency</i>	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	Bentuk rancangan sistem pengolah data mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem.
<i>Simplicity</i>	Kemudahan dalam pengelolaan, perbaikan, dan pengembangan sistem	Mudah untuk dikelola, diperbaiki, dan dikembangkan. Hal ini dapat dilihat pada tahapan-tahapan proses penulisan kode program.

d) Aspek *Usability*

Secara umum, definisi kebergunaan *usability* adalah derajat kemampuan sebuah perangkat lunak untuk membantu penggunanya dalam menyelesaikan sebuah tugas. Ada beberapa pakar yang memberikan definisi dan komponen kualitas dari kebergunaan, diantaranya:

1) Jacob Nielsen (2003):

Kebergunaan adalah atribut kualitas yang menunjukkan seberapa mudah suatu antarmuka digunakan. Terdiri atas lima komponen kualitas, yaitu: Mudah dipelajari (*learnability*), Efisiensi (*efficiency*), Mudah diingat (*memorability*), Kesalahan dan keamanan (*errors*), Kepuasan (*satisfaction*) (Nielsen's, hal. 1).

2) ISO 9241-11

Sejauh mana suatu produk dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam konteks tertentu penggunaan (Bevan, 2006).

Dari teori-teori di atas, pengujian dalam penelitian ini menggunakan lima aspek *usability* atau lima atribut kebenaran seperti yang di kemukakan oleh *Jacob Nielson* dengan menggunakan kuisisioner. Analisis *usability* dapat dilakukan menggunakan kuisisioner SUS (*System Usability Scale*) kuisisioner ini dikembangkan di *Digital Equipment Corp*, terdiri dari sepuluh pertanyaan. Setiap pertanyaan adalah pernyataan dan rating pada lima- titik skala "*Sangat tidak setuju*" hingga "*Sangat Setuju*" (Brooke, 1996). Untuk lebih jelasnya dapat dilihat pada Tabel 5.

Table 5. Instrumen *System Usability Scale* (SUS)

NO	Kuisoner
1	Saya berpikir untuk sering menggunakan website ini
2	Menurut saya website ini tidak terlalu kompleks
3	Menurut saya website ini mudah digunakan
4	Saya membutuhkan trainer untuk menggunakan wensite ini
5	Saya menemukan berbagai fungsi website ini terintegrasi dengan baik
6	Saya pikir ada inkonsistensi terlalu banyak di website ini
7	Menurut saya kebanyakan orang akan dengan mudah mempelajari penggunaan situs ini
8	Menurut saya website ini sangat rumit digunakan
9	Saya merasa sangat percaya pada website ini
10	Saya perlu belajar banyak tentang situs ini sebelum saya bisa menggunakannya dengan efektif

Dari kuisoner SUS sudah memberikan cara penghitungan dengan skor yang akan menghasilkan nilai skor 0-100. Pemberian nilai pada setiap item nilai kontribusi akan berkisar 0-4. Untuk item 1, 3, 5, 7 dan 9 kontribusi skor adalah posisi skala dikurangi 1. Untuk item 2, 4, 6, 8 dan 10, kontribusi adalah 5 minus posisi skala. Kemudian hasil dari kuisoner dikalikan 2.5 untuk mendapatkan nilai keseluruhan SUS (Brooke, 1996). Selanjutnya hasil dari pengujian kuisoner SUS akan dibandingkan dengan *range* nilai standar SUS (Brooke, 1996). Nilai standar

SUS juga memiliki 5 titik skala dari “Sangat Baik – Sangat Tidak Baik”. Seperti pada Tabel 6.

Table 6. *Standar System Usability Scale*

SUS Score	Nilai (GRADE)
0-60	F
60-70	D
70-80	C
80-90	B
90-100	A

e) Aspek *Efficiency*

Efficiency berhubungan dengan perilaku dan waktu yang dibutuhkan perangkat lunak, yang berkaitan dengan respon, waktu pemrosesan dan pemanfaatan sumber daya, yang mengacu pada sumber daya material (memori, CPU, koneksi jaringan) yang digunakan oleh perangkat lunak (Luis Olsina, Gustavo Rossi, 2001). *Efficiency* juga berkaitan dengan Sejauh mana perangkat lunak membuat penggunaan optimal dari sistem sumber daya seperti yang ditunjukkan oleh subatribut berikut: perilaku waktu, perilaku sumber daya (S.Pressman, 2001). Selain itu (Luis Olsina, Gustavo Rossi, 2001) juga memberikan pengertian *efficiency* dalam teorinya *Quality requirements tree* untuk *web application quality*, bahwa *efficiency* berhubungan dengan respon waktu kinerja, kecepatan *loading page* dan kecepatan *loading graphich*.

Pengujian aspek *Efficiency* dilakukan menggunakan aplikasi *YSloow* dan *webpagetest* yang secara spesifik menguji halaman *web* dari sisi *performance*.

1) *YSlow*

YSlow merupakan sebuah *addons* yang dapat berintegrasi hampir dengan semua *web browser*. *YSlow* bekerja dalam 3 tahap untuk mendapatkan hasilnya : (1) *YSlow* menjelajah DOM untuk menemukan semua komponen (gambar, script, *stylesheet*, dll) dalam halaman. Setelah menjelajahi *DOM*, *YSlow* *loop* melalui komponen Panel Bersih *Firebug* dan menambahkan mereka ke daftar komponen yang sudah ditemukan di *DOM*. (2) *YSlow* mendapat informasi tentang masing-masing komponen: ukuran, apakah itu *gzip*, *Expires header*, dll *YSlow* mendapat informasi ini dari *Panel Net Firebug* jika tersedia. Jika informasi komponen tidak tersedia dari *Panel Net* (misalnya, komponen ini dibaca dari *cache* atau memiliki respon 304) *YSlow* membuat *XML HttpRequest* untuk mengambil komponen dan melacak header dan informasi lainnya yang diperlukan. (3) *YSlow* mengambil semua data tentang halaman dan menghasilkan nilai untuk setiap aturan, yang menghasilkan nilai keseluruhan (Marcelduran, 2013).

2) *Webpagetest*

WebPagetest adalah sebuah proyek *opensource* yang terutama sedang dikembangkan dan didukung oleh Google sebagai bagian dari upaya kami untuk membuat web lebih cepat. *WebPagetest* adalah alat yang pada awalnya dikembangkan oleh AOL untuk digunakan secara internal dan *opensource* pada tahun 2008 di bawah lisensi BSD. Versi online di www.webpagetest.org dijalankan oleh WPO Yayasan untuk kepentingan masyarakat kinerja dengan

beberapa perusahaan dan individu yang menyediakan infrastruktur pengujian di seluruh dunia (google, 2008).

f) Aspek *Reliability*

Reliability merupakan *software quality* yang mengukur sejauh mana program dapat diharapkan untuk melakukan fungsi yang ditujukan dengan presisi yang diperlukan (S.Pressman, 2001). Sedangkan Menurut ANSI, *software reliability* adalah kemungkinan perangkat lunak bebas dari kesalahan operasi selama periode tertentu dalam suatu sistem tertentu. *Correctness* dan *reliability* mempunyai ketergantungan yang positif selama kesalahan pada perangkat lunak mengakibatkan kebenaran produk dan sistem yang *reliable* (Ginting, 2010). Dalam teorinya *Query requirements tree* (Luis Olsina, Gustavo Rossi, 2001) menjelaskan tentang *quality factors reliability* meliputi *Correct link processing*, *User input validation and recovery*, dan *Error recovery* (Olsina, Roger S. Pressman (5th ed.), 2001). *Correct link processing* merupakan pengujian pada semua *link web* dan memastikan semua *link* bekerja dengan benar sedangkan pengujian pada *User input validation and recovery*, dan *Error recovery* difokuskan pada pengujian *interface software* yang menampilkan validasi *input data* dan *error recovery* pada *pengimputan data*.

g) Aspek *Interoperability*

Seperti *quality factor* yang lain, interoperabilitas adalah sebuah konsep dengan banyak arti yang berbeda. Sebuah studi tentang definisi interoperabilitas ditemukan 22 arti yang berbeda (Folmer & Verhoosel). Definisi yang sering digunakan adalah dari IEEE: Interoperabilitas adalah kemampuan dari dua atau

lebih sistem atau komponen untuk bertukar informasi dan menggunakan informasi yang telah dipertukarkan (Legner & Lebreton, Rukanova, 2007). Definisi lain digunakan oleh Departemen Pertahanan AS (Tingkat Interoperabilitas Sistem Informasi): Kemampuan sistem, unit, atau kekuatan untuk menyediakan layanan dan menerima layanan dari sistem lain (Legner & Lebreton, Rukanova, 2007).

Pengujian aspek *interoperability* dilakukan menggunakan *cohesion metric* (LCC) dan *coupling metric* (CBO, RFC, NOC).

1) LCC (*Loose Class Cohesion*)

Menurut Biemen dan Kang LCC adalah jumlah *relative* pasangan *method* terhubung atau *transitif* terhubung. Di mana dua metode yang transitif terhubung jika mereka langsung atau tidak langsung terhubung ke sebuah *atribut* atau *variable* (Rakshith V, 2011). Menurut Bieen dan Kang menyatakan bahwa untuk $LCC = 1$ maka class *kohesif* sempurna dan untuk $LCC = 0$ maka *method* tidak terhubung dan sama sekali bukan *class kohesif*. LCC rendah menunjukkan bahwa sejumlah komponen kelas termasuk dalam kelas lain sebagai gantinya (F.Sumway, 1997). Penghitungan LCC dapat dilakukan dengan menggunakan *matric* hitung $LCC = (NDC + NID)/NP$, dengan NDC adalah jumlah koneksi langsung, NID jumlah koneksi tidak langsung, dan NP adalah jumlah koneksi maksimal dengan rumus $N*(N-1)/2$ dengan N adalah *method* atau *function*.

2) CBO (*Coupling Between Object Classes*)

Menurut Chidamber and Kemerer, 1994 CBO untuk kelas adalah hitungan jumlah kelas lain yang merupakan *coupled* atau pasangan. *Coupling* antara

dua kelas dikatakan terjadi ketika satu kelas menggunakan *method* atau *variabel* dari kelas lain. CBO diukur dengan menghitung jumlah kelas hierarki berbeda *non-heritance* (Rodriguez & Harrison, 2001).

Coupling Between Object Classes (CBO) juga dapat diartikan adalah hitungan untuk jumlah kelas yang saling berpasangan atau *coupe*. Hal ini diukur dengan menghitung kelas berpasangan, dengan kelas yang saling berpasangan bersifat *non-inheritance* satu sama lain (Rosenberg).

3) RFC

RFC adalah jumlah dari himpunan semua *method* yang dapat dipanggil dalam menanggapi pesan ke sebuah objek dari kelas atau oleh beberapa *method* dalam kelas. ini termasuk semua *method* diakses dalam hirarki kelas. RFC menghitung kejadian panggilan fungsi ke kelas-kelas lain dari kelas tertentu (Rodriguez & Harrison, 2001).

4) NOC

NOC adalah jumlah *subclass* langsung bawahan ke kelas dalam hierarki. NOC menghitung jumlah *subclass* milik kelas (Rodriguez & Harrison, 2001).

h) Aspek Portability

Menurut McCall, Richards, & Walters, 1977 aspek *portability* didefinisikan sebagai usaha yang diperlukan untuk mentransfer program dari satu perangkat keras dan / atau lingkungan sistem perangkat lunak yang lain (S.Pressman, 2001). Sedangkan menurut ISO 9126 *portability* didefinisikan sebagai kemudahan yang perangkat lunak dapat dialihkan dari satu. lingkungan yang lain seperti yang

ditunjukkan oleh *subattribute* berikut: beradaptasi, kemampuan, *installability*, kesesuaian, *replaceability* (S.Pressman, 2001).

Pengujian pada aspek *portability* dilakukan dengan melakukan *load tester* pada *web browser* ternama yang banyak digunakan baik dalam *browser* berbasis *desktop* maupun *browser* berbasis *mobile*. Untuk lebih jelasnya tentang instrumen *portability* dapat dilihat pada Tabel 7.

Table 7. Instrumen *Portability*

Aspek yang dinilai	Hasil yang diperoleh
Sistem dapat berjalan pada <i>browser</i> berbasis <i>desktop</i>	Hasil pengujian menunjukkan bahwa sistem kompatibel dengan beberapa <i>browser</i> ternama. Hal ini terbukti dari hasil pengujian bahwa sistem dapat di akses di beberapa <i>browser</i> seperti <i>Mozilla Firefox</i> , <i>Internet Explorer</i> , <i>Opera</i> , dan <i>Google Chrome</i> tanpa terdapat pesan <i>error</i> .
Sistem dapat berjalan pada <i>browser</i> berbasis <i>mobile</i>	Hasil pengujian menunjukkan bahwa sistem dapat diakses melalui <i>browser</i> berbasis <i>mobile</i> yaitu <i>Opera Mini</i> tanpa terdapat pesan <i>error</i> .

B. Penelitian yang relevan

Guna mendukung penelitian ini maka diperlukan penelitian lain yang relevan sebagai bahan perbandingan dan juga referensi, berikut penelitian yang sudah ada dan relevan dengan penelitian ini:

Analisis Sistem Informasi Pengelolaan Data Alumni Sekolah Berbasis *Codeigniter PHP Framework*, lokasi: SMA N 1 Yogyakarta, responden: 15 orang (Hanggara, 2012).

BAB III

METODE PENELITIAN

A. Desain Penelitian

1. Metode Penelitian

Melihat latar belakang dan tujuan, maka penelitian ini menggunakan pendekatan penelitian dan pengembangan atau *Research and Development* (R&D). Penelitian dan Pengembangan didefinisikan sebagai pekerjaan investigasi yang dilakukan untuk memperoleh pengetahuan untuk mengembangkan atau menerapkan produk baru atau prosedur, atau untuk membuat perbaikan yang signifikan untuk produk yang sudah ada (Earl A. Lund, 1993).

Sedangkan menurut Ali (2010) suatu proses dalam mengembangkan dan memvalidasi perangkat tertentu yang menjadi produknya, yang dalam respektif industri merupakan pengembangan satu *prototype* produk sebelum diproduksi secara massal. Dengan demikian, pengembangan lebih diarahkan pada upaya menghasilkan produk siap untuk digunakan secara nyata di lapangan, bukan hanya menemukan pengetahuan atau menguji hipotesis atau teori tertentu. Penelitian dan pengembangan yang dilakukan difokuskan pada pengembangan perangkat lunak sistem informasi untuk pengelolaan data Anggota pada UKM Musik Sisma UNY.

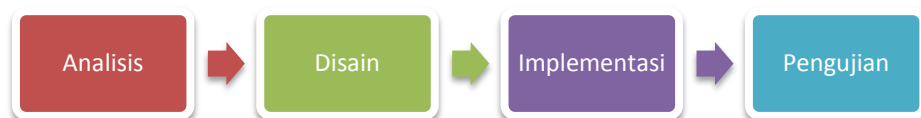
2. Tempat dan Waktu Penelitian

Waktu penelitian diambil sekitar bulan Juli 2013 sampai Oktober 2013 dengan pertimbangan sistem siap diimplementasikan dan dilakukan pengujian.

Tempat penelitian dilakukan di UKM Band Sigma Universitas Negeri Yogyakarta yang terletak di Student Center Lt1 sayap barat.

B. Alur Penelitian

Agar produk yang dihasilkan dalam pengembangan sesuai dengan tujuan, maka penelitian ini menggunakan pendekatan model pengembangan rekayasa perangkat lunak *Waterfall*, di mana produk melalui beberapa tahapan, yaitu tahap analisis kebutuhan (*requirement*), tahap desain, tahap implementasi, dan terakhir tahap pengujian aplikasi sebelum akhirnya siap untuk digunakan (A.S & Shalahuddin, 2013).



Gambar 2. Waterfall
(Langkah Penggunaan Metode *Waterfall* (Wiswakarma, 2010)).

Karakteristik dari model *waterfall* adalah urutan pengembangan yang menekankan perencanaan pada tahap awal, memastikan desain awal atau spesifikasi awal sistem dalam pengembangan sesuai dengan yang dibutuhkan, Selain itu, dokumen dan perencanaan intensif membuatnya bekerja dengan baik dalam pengembangan proyek (Mohammed & Munassar, 2010, hal. 95).

1. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengetahui kebutuhan serta masalah-masalah yang perlu diselesaikan. Dibutuhkan berbagai sumber informasi

mengenai kasus dan kebutuhan yang dibutuhkan oleh pengguna, di antaranya dari pihak UKM yang bersangkutan. Kemudian dibuat daftar permintaan atau kebutuhan pengguna (*user requirement list*) yang perlu disediakan pada aplikasi sistem informasi pengelolaan data UKM yang dikembangkan dalam penelitian ini.

2. Desain

Berdasarkan dari analisis kebutuhan maka dapat diketahui apa saja yang menjadi kebutuhan dari pengembangan sistem informasi pengelolaan data anggota UKM, sehingga sistem yang dibuat nantinya sesuai dengan apa yang diharapkan. Tahap desain meliputi:

a) Perancangan Unified Modeling Language (UML)

Perancangan cara kerja program menggunakan UML yang meliputi desain *Use Case Diagram* dan *Activity Diagram*

b) Perancangan antar muka pengguna (User Interface)

Rancangan interfaces dari aplikasi *web* ini yang menggunakan CSS terintegrasi dengan HTML.

c) Perancangan desain basis data.

Mendesain tabel-tabel data yang dibutuhkan serta menjelaskan relasi antar tabel.

3. Tahap Implementasi

Implementasi merupakan tahap saat pembuatan aplikasi dimulai setelah dilakukan analisis dan desain. Rancangan program yang telah disiapkan kemudian

implementasikan dalam bahasa pemrograman, sehingga semua fungsi dapat dijalankan dengan baik oleh pengguna.

4. Tahap pengujian

Pada tahap ini aplikasi yang telah dikembangkan kemudian diberikan berbagai rangkaian pengujian kualitas perangkat lunak yang menggunakan beberapa instrumen penelitian sesuai Standar Ahli, sehingga dapat dilakukan evaluasi sistem sebelum akhirnya dapat digunakan oleh banyak pengguna. Untuk mengetahui kualitas perangkat lunak yang dikembangkan dalam penelitian ini, perangkat lunak diuji dan dianalisis menggunakan kualitas perangkat lunak diantaranya *Correctness*, *Maintainability*, *Integrity*, *Usability*, *Efficiency*, *Reusability*, *Interoperability* dan *Portability*. Pengujian yang dilakukan dengan *load tester* atau software dilakukan secara berulang ulang guna mendapatkan hasil yang valid. Sedangkan pengujian *black-box* (aspek *integrity* dan *correctness*) testing sebagai pendukung pengujian dilakukan sendiri oleh pengembang.

a) Pengujian *Integrity*

Pengujian *Integrity* difokuskan pada kemampuan *software* untuk memberikan batasan hak akses bagi *user* atau mencegah *user* melakukan akses yang tidak sah secara sengaja maupun tidak sengaja. Pengujian akan dilakukan menggunakan kusioner dan *tools* khusus berupa *software* yang akan menguji *software* dari sisi *security*.

b) Pengujian *Usability*

Pengujian *Usability* dilakukan untuk menilai seberapa mudah antar muka pengguna pada *software* yang berkaitan dengan kemudahan penggunaan, efisiensi, *error* serta kepuasan *user* dalam menggunakan *software*. Pengujian ini dilakukan menggunakan angket berupa kuisioner yang di isi oleh 30 responden dari UKM Band Sicma UNY.

c) Pengujian *Maintainability*

Pengujian aspek *maintainability* dapat dilakukan dengan melakukan pengujian pada aspek *instrumentation*, *consistency*, dan *simplicity*.

d) Pengujian *Correctness*

Faktor kualitas *correctness* dianalisa dengan menghitung jumlah *error* tiap kilo lines of code (KLOC). Kemudian hasil penghitungan dibandingkan dengan teori *McConnel* tentang batas toleransi *error* berdasarkan jumlah *KLOC*

e) Pengujian *Efficiency*

Pengujian pada aspek *efficiency* merujuk pada performa dari sistem informasi yang dikembangkan. Pengujian dilakukan menggunakan *software* dilakukan secara *load tester*.

f) Pengujian *Reliability*

Pengujian *reliability* memfokuskan pada *Correct link processing*, *User input validation and recovery*, dan *Error recovery* sesuai dengan teori olsina (Luis Olsina, Gustavo Rossi, 2001) *Query recuirements tree* tentang *reliability*.

Pengujian *Correct link processing* dilakukan menggunakan *software Xenu Link Sleuth 1.3.8*. dan dilakukan dengan cara *load tester*, sedangkan pengujian *User input validation and recovery*, dan *Error recovery* dilakukan secara *load tester* pada *browser*.

g) Pengujian *Interoperability*

Pengujian aspek *interoperability* dilakukan menggunakan *cohesion metric* LCC dan *coupling metric* CBO, RFC and NOC (Saradhi, 2010).

5. Instrumen Penelitian

Instrumen penelitian terdiri dari angket dan tools yang digunakan sebagai Pengujian terhadap aspek *Correctness*, *Maintainability*, *Integrity*, *Usability*, *Efficiency*, *Reliability*, *Interoperability* dan *Portability*.

a) Instrumen *Correctness*

SLOC-Sourch Line of Code merupakan *software opensourch* yang digunakan untuk menghitung jumlah baris kode dalam sebuah peroject *software* terutama aplikasi web dengan lisensi dari *Cuong Tham* pada situs resminya (Tham, 2007).

b) Instrumen *Usability*

Instrumen yang digunakan pada aspek *usability* menggunakan kuisioner SUS (*System Usability Scale*).

c) Instrumen *Integrity*

Instumen yang digunakan pada pengujian aspek *integrity* menggunakan *Acunetix Web Vulnerability Scanner* dan kuisioner.

d) Instrumen *efficiency*

Pengujian aspek *efficiency* dilakukan menggunakan aplikasi *YSloow* dan *webpagetest* yang secara spesifik menguji halaman *web* dari sisi *performance*.

e) Instrumen *Reliability*

Instrumen pengujian *relibility* dilakukan menggunakan aplikasi *Xenu Link Sleuth* dan metrik.

f) Instrumen *Maintainability*

Pengujian *maintainability* menggunakan ukuran-ukuran (*metrics*) yang kemudian pengujian dilakukan uji secara operasional (Land, 2002).

g) Instrumen *Interoperability*

Pengujian aspek *interoperability* dilakukan menggunakan *cohesion metric* (LCC) dan *coupling metric* (CBO, RFC, NOC).

h) Instrumen *Portability*

Pengujian pada aspek *portability* dilakukan dengan melakukan *load tester* pada *web browser* ternama yang banyak digunakan baik dalam *browser* berbasis *dekstop*.

BAB IV

HASIL DAN PEMBAHASAN

A. Tahap Analisis Kebutuhan

1. Analisis Proses

Beberapa fungsi minimal yang dibutuhkan antara lain:

- a) Anggota UKM dapat mendaftarkan data diri, menyunting, mencari data secara mandiri kapan saja dan dari mana saja (tidak harus datang ke sekretariat UKM).
- b) Anggota UKM dapat mencari data sesama anggota UKM lainnya.
- c) Pengurus UKM dapat menambah, menyunting, menghapus data anggota UKM
- d) Anggota UKM dapat melihat statistik data anggota (*study tracer*).
- e) Pengurus UKM dapat mencetak data anggota.
- f) Data anggota UKM harus dijaga keamanannya karena sangat sensitif.

2. Analisis Kebutuhan *Hardware* dan *Software*

Dari analisa sistem kemungkinan kepadatan dan besar data, dapat ditentukan kapasitas *hardware* yang memenuhi syarat yang digunakan. Secara umum hardware yang digunakan dalam sistem ini adalah :

- a) Untuk *server*, satu unit komputer *server* sebagai *server* yang telah diinstall dan dikonfigurasi sesuai standar minimal, yaitu *Apache Web Server*, PHP, dan sistem basis data MySQL terkoneksi internet.

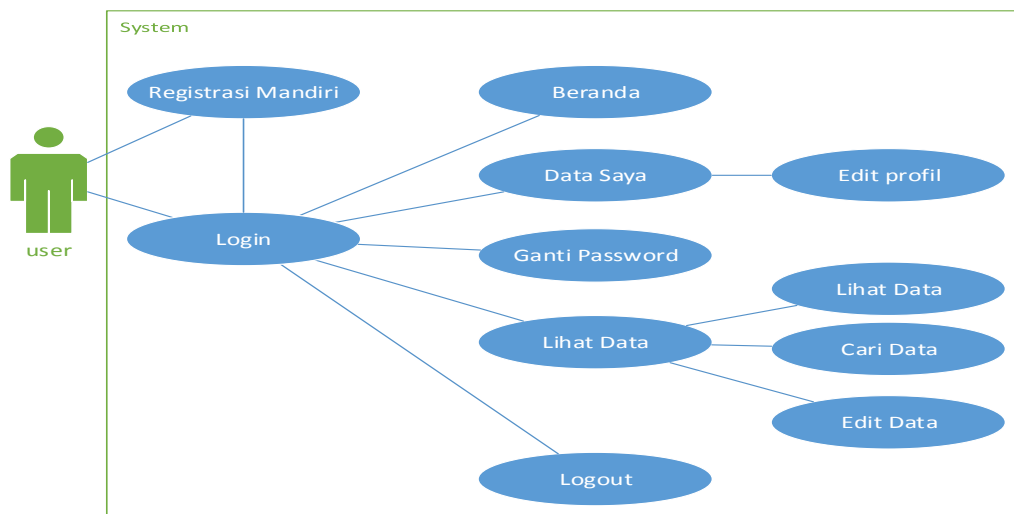
- b) Untuk pengguna, harus bisa diakses dari komputer dan laptop, serta aplikasi *Web browser* pada umumnya (spesifikasi minimum).

B. Tahap Desain

1. Perancangan Unified Modelling Language (UML)

a) *Use Case Diagram* untuk sistem yang dikembangkan

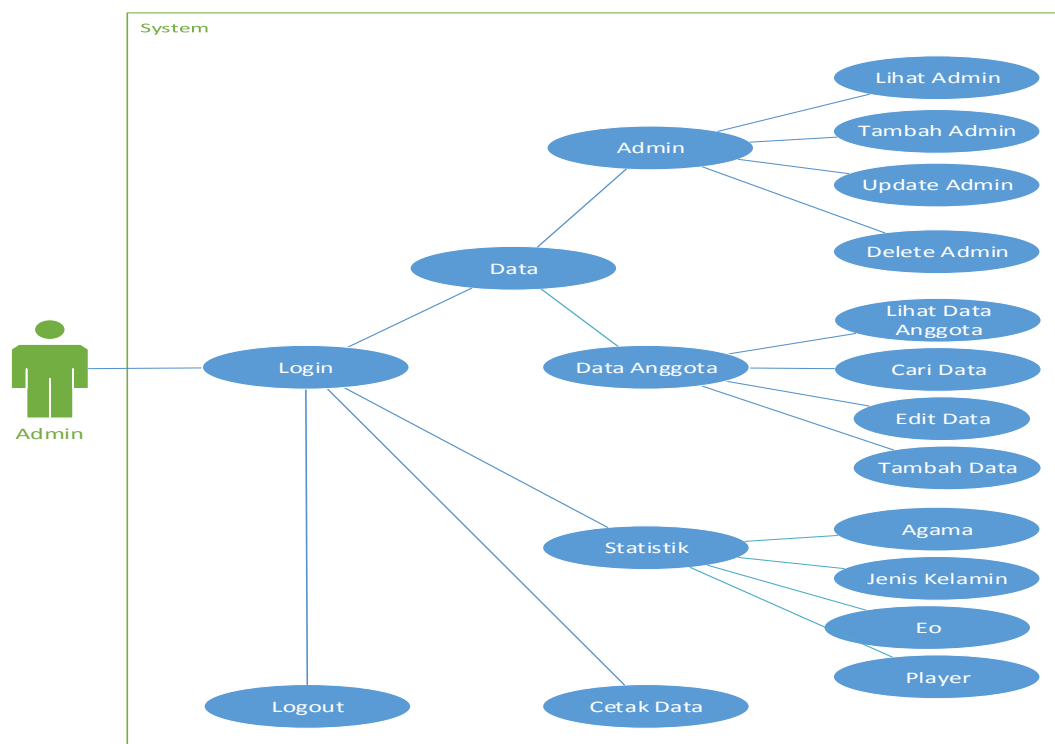
Use Case Diagram terdiri dari sebuah aktor dan interaksi yang dilakukannya, aktor tersebut berinteraksi langsung dengan sistem. Pada pengembangan perangkat lunak, *Use Case Diagram* menjelaskan tentang hubungan antara sistem dengan aktor. Hubungan ini dapat berupa *input* ke sistem ataupun *output* ke aktor. Untuk lebih jelasnya *use case diagram admin* dapat dilihat pada Gambar 3.



Gambar 3. *Use Case User*

Pada *Use Case Diagram* tersebut menunjukkan bahwa aktor dalam hal ini *user* dapat melakukan interaksi dengan *system*. Pada *system*, *user* dapat melakukan pendaftaran anggota secara mandiri, *login* setelah melakukan registrasi dan memiliki *username* dan *password* untuk masuk kedalam *system*, *user* dapat melakukan sunting data pribadi, melakukan pencarian data anggota, dan melihat data anggota yang lain (dengan hak terbatas).

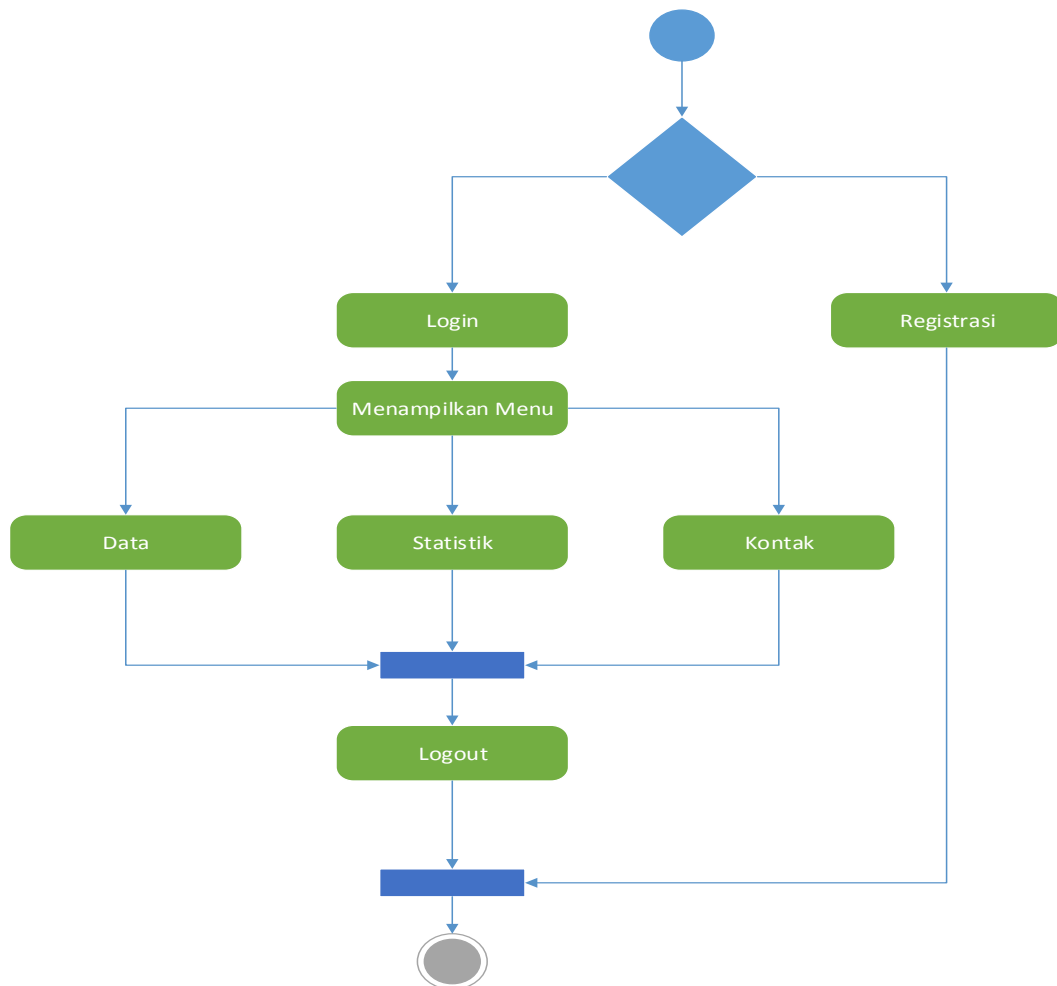
Demikian juga dengan *use case diagram admin*, hanya saja pada *admin* terdapat perbedaan hak akses yang lebih luas dari pada *user*. Hal ini dikarenakan *admin* memiliki hak akses penuh pada perubahan data *user*. Untuk lebih jelasnya *use case diagram admin* dapat dilihat pada Gambar 4.



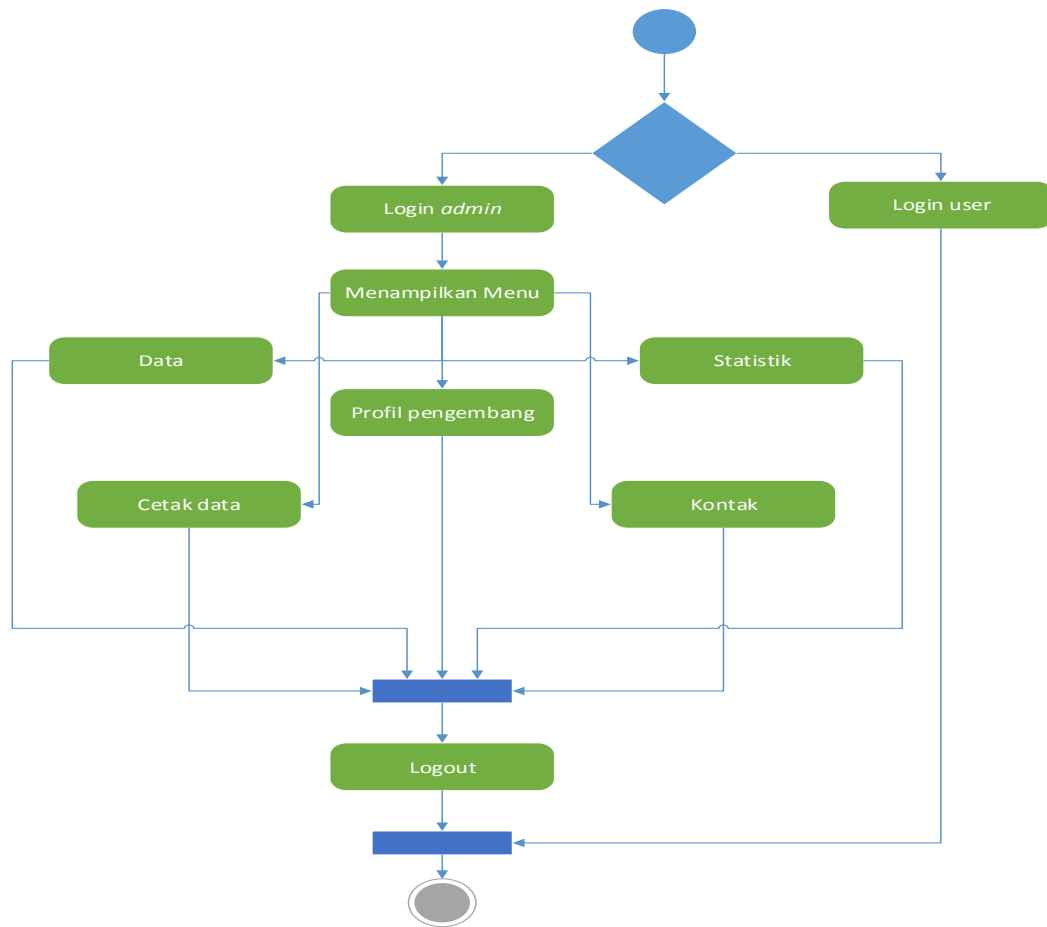
Gambar 4. *Use Case Admin*

b) Perancangan *Activity diagram*(Aktivitas diagram)

Diagram aktifitas atau *activity diagram* menggambarkan *workflow* atau aliran kerja atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Untuk lebih jelasnya aktivitas diagram sistem informasi pendataan anggota UKM Band Sicma UNY dapat dilihat pada Gambar 5 untuk *activity diagram* user dan Gambar 6 untuk *actifity diagram* admin.



Gambar 5. *Actifity Diagram User*

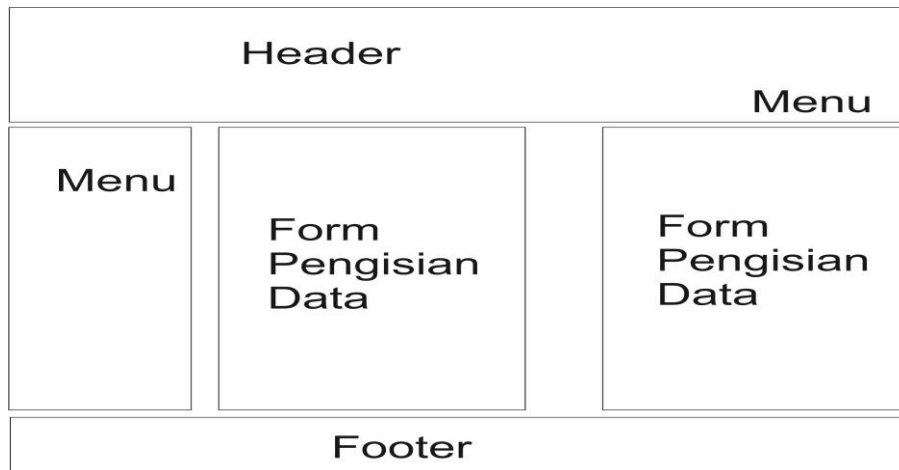


Gambar 6. *Actifity Diagram Admin*

2. Perancangan user interface (Antar Muka Pengguna)

a) Halaman Login user

Halaman *login* yang akan dikembangkan berisi *form username* dan *password* seperti Gambar 7.

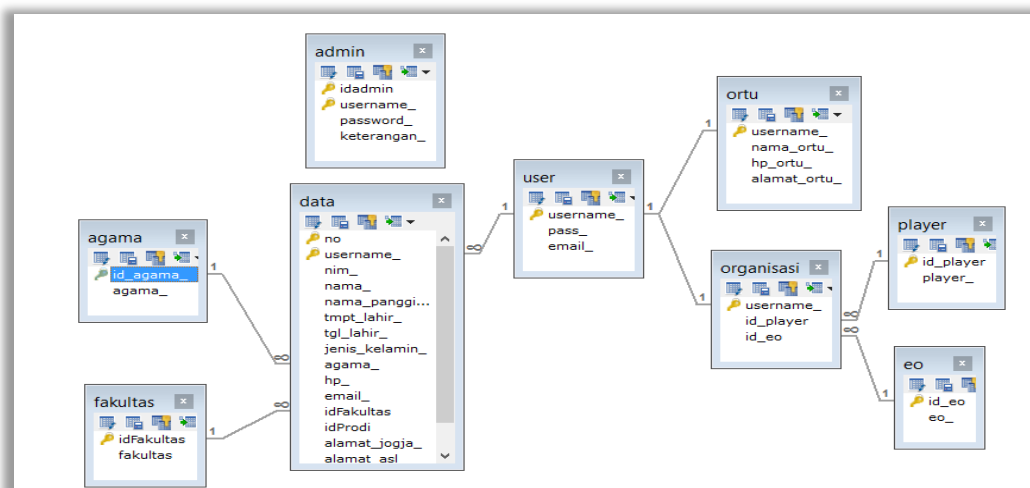


Gambar 7. Perancangan *User Interface* Form Pengisian Data

Untuk perancangan *user interface* lain dapat dilihat pada lampiran

3. Perancangan Basis Data

Perancangan basis data berisi tentang perencanaan *database* yang akan digunakan pada perangkat lunak. Pada *database* ini akan ada beberapa tabel sebagai penyimpanan data di antaranya : *admin*, *agama*, *data*, *eo*, *fakultas*, *organisasi*, *ortu*, *player*, *user*. Untuk lebih jelasnya dapat dilihat pada Gambar 8.



Gambar 8. Rancangan Desain Basis Data

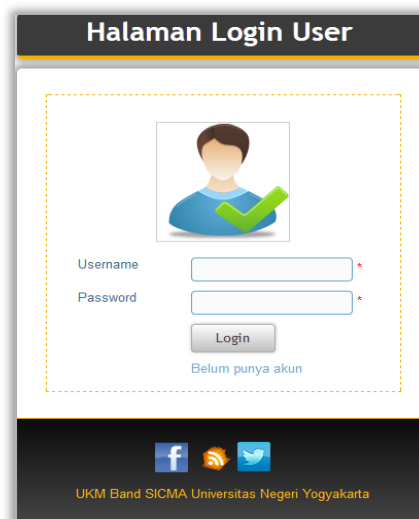
C. Tahap Implementasi

Implementasi merupakan tahap pembuatan aplikasi setelah melalui tahap analisis kebutuhan dan desain. Rancangan program yang telah disiapkan kemudian di terapkan dalam bahasa pemrograman sehingga menghasilkan aplikasi dengan fungsi yang berjalan sesuai dengan yang diinginkan.

1. Implementasi *User Interface* (Antar Muka Pengguna)

a) Halaman Login

Halaman *login* berisi *form username* dan *password user* dan sebuah *link* registrasi mandiri di bawah tombol *login*. Untuk lebih jelas lihat Gambar 9.



Gambar 9. Form Login User

b) Registrasi Mandiri

Halaman registrasi mandiri berisi semua *form input* yang dibutuhkan sistem. Untuk lebih jelas lihat Gambar 10.

PENDATAAN ANGGOTA UKM BAND SICMA
UNIVERSITAS NEGERI YOGYAKARTA

ACCOUNT

FORM REGISTRASI DATA ANGGOTA

INPUT DATA USER

Username

Pass

Pass.conf

INPUT DATA ORGANSATION

Role

DO

INPUT DATA PRIBADI

NIM

Nama Lengkap

Panggilan

Tanggal lahir

Jenis kelamin ☐ Laki-laki ☐ Perempuan

Agama

Hp

Email

Fakultas

Prodi

Alamat (Jalan, RT/RW, Desa/Kelurahan)

Tahun masuk

Simpan

Gambar 10. Form Registrasi Mandiri

c) Halaman Profil

Halaman profil berisi informasi data pribadi setelah *login* seperti pada Gambar

11.

PENDATAAN ANGGOTA UKM BAND SICMA
UNIVERSITAS NEGERI YOGYAKARTA

MENU
Contact
Data
Status
Logout

Data Anggota "gondil"

DATA PROFIL

Nim : 08520241002
Nama : ridho faris primastomo
Nama panggilan : gondil
Tempat lahir : sleman
Tgl. lahir : 1999-10-10
Jenis kelamin : Laki-laki
Agama : Islam
Hp : 08562894489
Email : sleeping.anthem@gmail.com
Fakultas : TEKNIK
Prodi : Pendidikan Teknik Informatika
Alamat yogya : cibukan sumberadi
Alamat asal : cibukan sumberadi mlati sleman
Tahun masuk : 1997

DATA ORGANISASI ANDA

Player : Gitaris
EO : Event Organizer

DATA ORANGTUA ANDA

Nama Orangtua : sujarno add
Hp Orangtua : 123435455343
Alamat Orangtua : cibukan sumberadi

Untuk Mengubah data diri silahkan klik link disini

Copyright_Jun © 2011-2012 Facebook | Twitter | Blog

Gambar 11. Form Lihat Profil

d) Halaman Edit Profil

Halaman edit profil hampir sama dengan halaman registrasi mandiri. Hanya saja sudah terisi data diri sesuai dengan *input* data. Untuk lebih jelas lihat Gambar 12.

PENDATAAN ANGGOTA UKM BAND SICMA
UNIVERSITAS NEGERI YOGYAKARTA

MENU
Data
Status
Contact

Form Edit Profil

INPUT DATA USER

Username : faris
Pass : *****
Passconf : *****

INPUT DATA ORGANISATION

Player : Gitaris
EO : Event Organizer

INPUT DATA ORANGTUA / WALI

Nama Orangtua / Wali : sujarno add
Hp : 19916191619
Alamat : cibukan sumberadi

INPUT DATA PRIBADI

Nim : 08520241002
Nama : ridho faris primastomo
Panggilan : gondil ganteng maksimal
Tempat lahir : sleman
Tanggal lahir : 31/10/1999
Kelamin : ☒ Laki-laki ☐ Perempuan
Agama : Islam
Hp : 08562894489
Email : sleeping.anthem@gmail.com
Fakultas : TEKNIK
Prodi : Pendidikan Teknik Informatika
Alamat jogja : sleman
Alamat asal : cibukan sumberadi mlati sleman
Tahun masuk : 2008

save Cancel

Copyright_Jun © 2011-2012

Gambar 12. Form Edit Profil

e) Halaman Cari dan Lihat Data

Halaman lihat dan cari data berisi semua data anggota yang terdaftar dan terdapat form pencarian di kanan atas halaman. Untuk lebih jelas lihat Gambar 13.



The screenshot shows a web application interface for 'PENDATAAN ANGGOTA UKM BAND SICMA' at 'UNIVERSITAS NEGERI YOGYAKARTA'. It features a sidebar menu with 'Data', 'Statistik', and 'Contact'. The main area displays a table of members with columns for No, Nim, Nama, Nama Panggilan, CP, Email, and Action. A search bar is located at the top right of the table area.

No	Nim	Nama	Nama Panggilan	CP	Email	Action
1	08520241002	ridho faris primastomo	gondil ganteng maksimal	08562894489	sleeping.anthem@gmail.com	Detail
2	0877393486598	chisnul liyna	liyna	85643789022	liyna@gmail.com	Detail
3	8	bowol	bowo slak	087484983039	bowo@gmail.com	Detail
4	097987897	ari pujianto	sayuti	8739389290	ari_p@gmail.com	Detail
5	07518244001	Taufiq Fada Ardena	Taufiq	081804170965	taufiqfada@ymail.com	Detail
6	08765443290	sicma	sicma	098765432	sicma@sicma.com	Detail
7	08987652739	sicmajunior	sicmajunior	0897867564	sicmajunior@sicmajunior.com	Detail
8	08520241004	Yoga hanggara	yoga	08763872948	yoga@gmail.com	Detail
9	08520241022	Muhamad Rizki A.W	Rizki	085269947968	rizqira@gmail.com	Detail

Gambar 13. Form Cari-Lihat Data Anggota

f) Halaman Detail Data Anggota

Halaman ini menampilkan detail data anggota lain dan dapat dilihat sesama *user*. Untuk lebih jelas lihat Gambar 14.



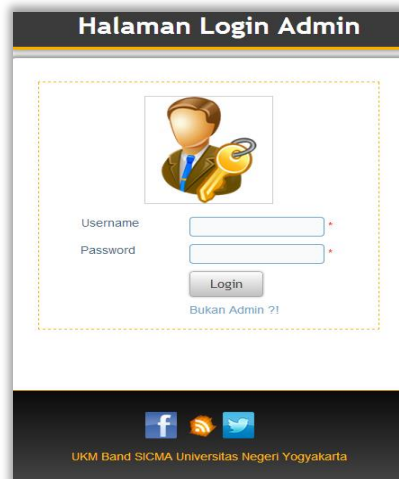
The screenshot shows the same web application interface, but the main area displays the detailed information for a specific member, identified by the search term 'liyna'.

Data Anggota "liyna"	
Nim	: 0877393486598
Nama	: chisnul liyna
Nama panggilan	: liyna
Tempat lahir	: sleman
Tgl. lahir	: 1970-05-04
Jenis kelamin	: Perempuan
Agama	: Islam
Hp	: 85643789022
Email	: liyna@gmail.com
Fakultas	: BAHASA DAN SENI
Prodi	: Pendidikan Bahasa Inggris
Alamat yogya	: Cibukan sumberadi mlati sleman yogyakarta
Alamat asal	: Cibukan sumberadi mlati sleman yogyakarta
Tahun masuk	: 2013

Gambar 14. Form Detail Data Anggota

g) Login Admin

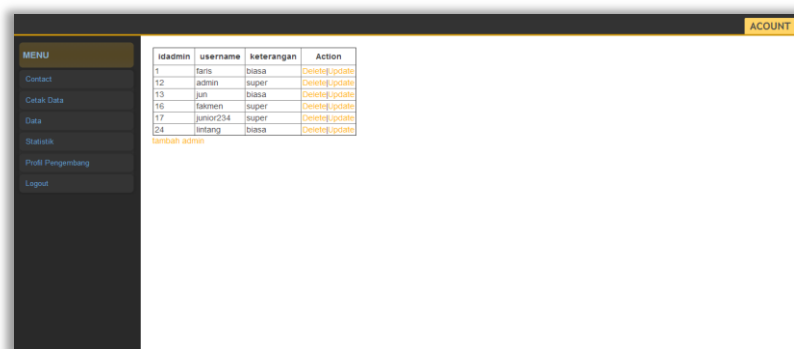
Halaman *login admin* berisi *username* dan *password* dengan *link bukan admin* di bawah tombol *login* yang akan membawa *user* ke halaman *login user*. Untuk lebih jelas lihat Gambar 15.



Gambar 15. Form Login Admin

h) Halaman Data Admin

Halaman ini menampilkan jumlah seluruh *admin* dan status *admin*. Untuk lebih jelas lihat Gambar 16.



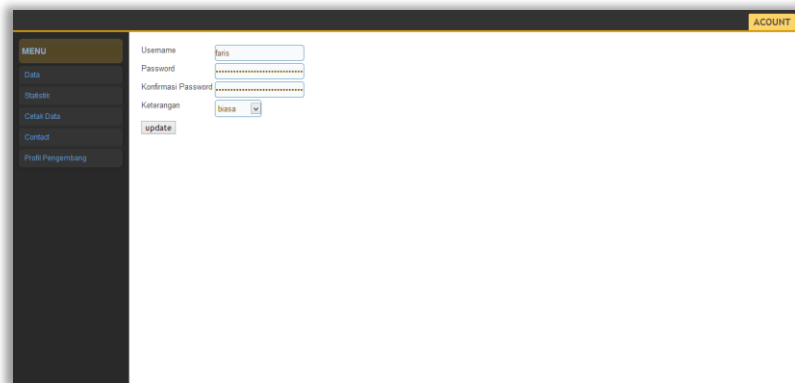
idadmin	username	keterangan	Action
1	faris	biasa	Delete update
12	admin	super	Delete update
13	jun	biasa	Delete update
16	lakmen	super	Delete update
17	junior234	super	Delete update
24	lintang	biasa	Delete update

[tambah admin](#)

Gambar 16. Form Lihat Admin

i) Halaman Admin – Edit Admin

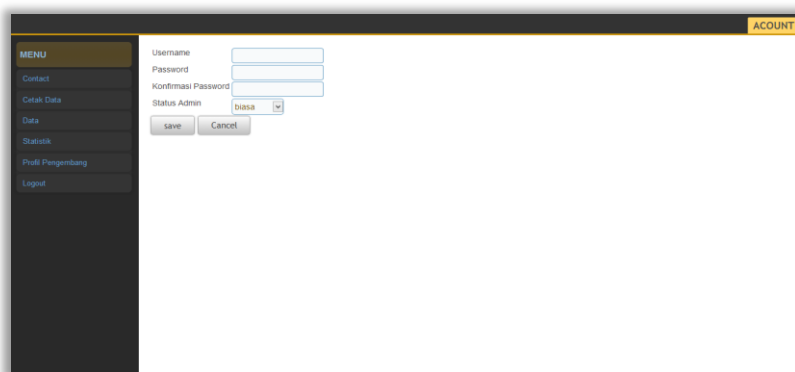
Halaman ini berisi *form* untuk mengubah data *admin*. Untuk lebih jelasnya lihat Gambar 17.



Gambar 17. Form Update Admin

j) Halaman Admin – tambah admin

Halaman tambah *admin* berisi *form* *username*, *password*, konformasi *password* dan status. Untuk lebih jelas lihat Gambar 18.



Gambar 18. Form Tambah Admin

k) Halaman admin – view data anggota

Halaman *view data anggota – admin*, menampilkan seluruh jumlah anggota yang terdaftar dengan fungsi yang dapat diakses : *detail, delete, update* seperti pada Gambar 19.

No	Nim	Nama	Nama Panggilan	CP	Action
1	08520241002	Indro Faria primastomo	gondri	0852294489	Detail Delete Update
2	0877293486598	ichsanul lyra	lyna	85643789622	Detail Delete Update
3	8	Ismael	Ismael	0874349836299	Detail Delete Update
4	097967897	ari putanto	isaydi	8739395296	Detail Delete Update
5	07518244001	Taufiq Fada Andena	Taufiq	081884178965	Detail Delete Update
6	08765443290	sicma	sicma	098765432	Detail Delete Update
7	08987652379	icmaganur	icmaganur	0897667564	Detail Delete Update
8	08520241004	Yoga hanggara	yoga	08763872948	Detail Delete Update
9	08520241022	Muhamad Rizki A.W	Rizki	085269947968	Detail Delete Update
10	08520241001	jati yehuar	joniari	09014639267	Detail Delete Update
11	09506134024	PIANCU	PIANCU	085624420227	Detail Delete Update
12	10207244011	Dandi Hilmi	Acong	085749310003	Detail Delete Update
13	10209244050	azizah hakim nur laila	jaahh	085729559555	Detail Delete Update
14	10202619111	REITO SUKAWARJO	OTONAK	085642081513	Detail Delete Update
15	11206244036	ginsar sidik	anjari	08985066961	Detail Delete Update
16	06201244069	Bagus Ady Kumawan	Bag	+62 867 2900 60	Detail Delete Update
17	09412144018	ryan gah syahada	ryan	085643789796	Detail Delete Update
18	11412141040	Muhammad Saleh A	Saleh	08567213355	Detail Delete Update
19	12501244018	Julian Rinjani Putra	Julian	085668141854	Detail Delete Update
20	09301241036	Yulan Angga Pratwi	Twi	085725303037	Detail Delete Update

Gambar 19. Form Lihat dan Cari Data (Admin)

l) Halaman Admin – tambah data anggota

Pada halaman ini, halaman yang digunakan sama dengan jika *user* melakukan pendaftaran secara mandiri, hanya saja terdapat privilege yang berjalan sehingga jika yang melakukan admin akan *diredirect* ke halaman *view data all*, namun jika *user* yang melakukan pendaftaran mandiri akan di *redirect* ke halaman *login*.

INPUT DATA USER

Username:

Password:

Name:

Name Panggilan:

Panggilan:

Tanggal lahir:

Tanggal lahir:

Agama:

Hp:

Email:

Fakultas:

Prodi:

Alamat jalan:

Alamat email:

INPUT DATA ORGANISATION

Payer:

EO:

INPUT DATA ORANGTUA / WALI

Nama Orangtua Wali:

Hp:

Alamat:

Save Cancel

Gambar 20. Form Tambah Data (Admin)

2. Implementasi Basis Data

Setelah membuat desain *database*, selanjutnya desain dikembangkan ke dalam bentuk *query* MySQL sehingga terbentuk tabel-tabel basis data yang dibutuhkan dalam pengembangan sistem informasi.

a) Tabel. data admin

Pada tabel *admin* berisi *idadmin*, *username*, *password* dan keterangan. Untuk lebih jelasnya lihat Gambar 21.

#	Kolom	Jenis	Penyortiran	Atribut	Kosong	Default	Ekstra	Aksi
1	<u>idadmin</u>	int(11)		Tidak	None	AUTO_INCREMENT		Ubah Hapus Selebihnya
2	username	varchar(45)	latin1_swedish_ci	Tidak	None			Ubah Hapus Selebihnya
3	password	varchar(45)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
4	keterangan	varchar(7)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya

Gambar 21. Implementasi Tabel Admin

b) Tabel data data

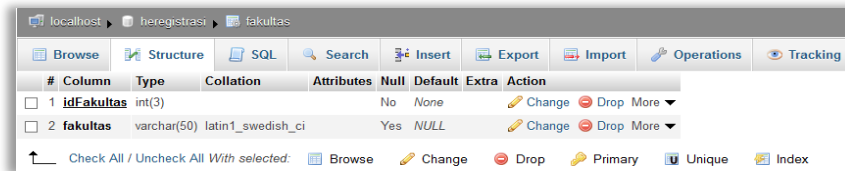
Implementasi dari tabel data berisi seluruh kolom data yang dibutuhkan dalam pendataan anggota UKM. Untuk lebih jelasnya lihat Gambar 22.

#	Kolom	Jenis	Penyortiran	Atribut	Kosong	Default	Ekstra	Aksi
1	<u>id</u>	int(11)		Tidak	None	AUTO_INCREMENT		Ubah Hapus Selebihnya
2	username	varchar(45)	latin1_swedish_ci	Tidak	None			Ubah Hapus Selebihnya
3	nim	varbinary(15)		Ya	NULL			Ubah Hapus Selebihnya
4	nama	varchar(45)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
5	nama_panggilan	varchar(45)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
6	tmpt_lahir	varchar(45)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
7	tgl_lahir	date		Ya	NULL			Ubah Hapus Selebihnya
8	jenis_kelamin	enum('L', 'P')	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
9	hp	varbinary(15)		Ya	NULL			Ubah Hapus Selebihnya
10	email	varchar(45)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
11	idFakultas	int(3)		Ya	NULL			Ubah Hapus Selebihnya
12	idProdi	int(5)		Ya	NULL			Ubah Hapus Selebihnya
13	alamat_jogja	varchar(100)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
14	alamat_asl	varchar(100)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Selebihnya
15	tahun_masuk	char(11)	latin1_swedish_ci	Tidak	None			Ubah Hapus Selebihnya

Gambar 22. Implementasi Tabel Data

c) Tabel fakultas

Tabel fakultas berisi informasi fakultas yakni *idfakultas* dan fakultas. Untuk lebih jelas lihat Gambar 23.



The screenshot shows the MySQL Structure window for the 'fakultas' table. The table has two columns: 'idfakultas' (int(3)) and 'fakultas' (varchar(50) with latin1_swedish_ci collation). The 'idfakultas' column is not null and has no default value. The 'fakultas' column is not null and has a default value of NULL. The 'fakultas' column is marked as a primary key.

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	idfakultas	int(3)			No	None		Change Drop More
2	fakultas	varchar(50)	latin1_swedish_ci		Yes	NULL		Change Drop More

Gambar 23. Implementasi Tabel Fakultas

d) Tabel user

Tabel *user* berisi *username* dan *password* yang digunakan pada saat *login* ke sistem. Untuk lebih jelas lihat Gambar 24.



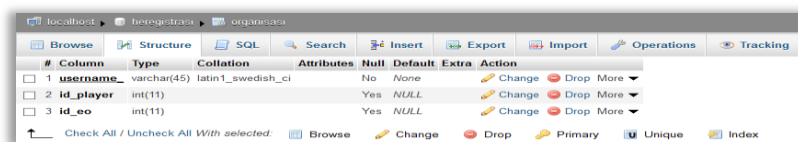
The screenshot shows the MySQL Structure window for the 'user' table. The table has three columns: 'username' (varchar(45) with latin1_swedish_ci collation), 'password' (varchar(45) with latin1_swedish_ci collation), and 'email' (varchar(45) with latin1_swedish_ci collation). The 'username' and 'password' columns are not null and have no default values. The 'email' column is not null and has a default value of NULL. The 'email' column is marked as a primary key.

#	Kolom	Jenis	Penyortiran	Atribut	Kosong	Default	Ekstra	Aksi
1	username	varchar(45)	latin1_swedish_ci		Tidak	None		Ubah Hapus Selebihnya
2	pass	varchar(45)	latin1_swedish_ci		Tidak	None		Ubah Hapus Selebihnya
3	email	varchar(45)	latin1_swedish_ci		Ya	NULL		Ubah Hapus Selebihnya

Gambar 24. Implementasi Tabel User

e) Tabel Organisasi

Tabel organisasi merupakan tabel primer dari *player* dan *eo*. Untuk lebih jelasnya lihat Gambar 25.



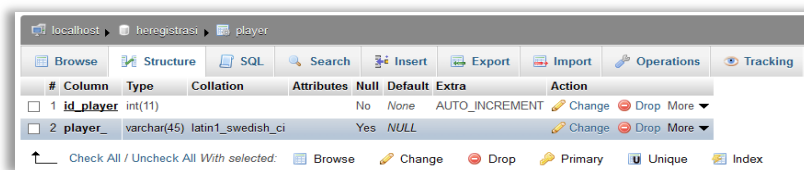
The screenshot shows the MySQL Structure window for the 'organisasi' table. The table has three columns: 'username' (varchar(45) with latin1_swedish_ci collation), 'id_player' (int(11)), and 'id_eo' (int(11)). The 'username' column is not null and has no default value. The 'id_player' and 'id_eo' columns are not null and have no default values. The 'id_eo' column is marked as a primary key.

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	username	varchar(45)	latin1_swedish_ci		No	None		Change Drop More
2	id_player	int(11)			Yes	NULL		Change Drop More
3	id_eo	int(11)			Yes	NULL		Change Drop More

Gambar 25. Implementasi Tabel Organisasi

f) Tabel Player

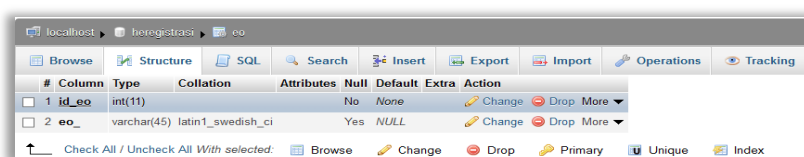
Tabel *player* bersisi seluruh data *skill player* yang dibutuhkan sebagai bahan pilihan dalam pengisian data. Untuk lebih jelasnya lihat gambar 26.



Gambar 26. Implementasi Tabel Player

g) Table Eo

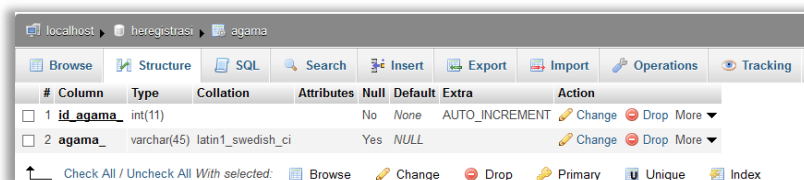
Tabel *Eo* berisi data pilihan *Eo* yang dibutuhkan sebagai bahan pilihan dalam pengisian data. Untuk lebih jelasnya lihat Gambar 27.



Gambar 27. Implementasi Tabel Eo

h) Tabel Agama

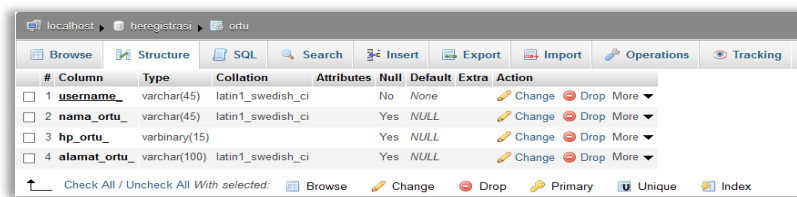
Tabel agama berisi *id_agama* sebagai *primary key* dan *agama_* yang berisi pilihan agama sebagai pilihan bagi *user*. Untuk lebih jelasnya lihat Gambar 28.



Gambar 28. Implementasi Tabel Agama

i) Tabel Ortu

Tabel *Ortu* merupakan tabel yang digunakan untuk menampung data orang tua anggota dengan kolom *username_* sebagai *index*. Untuk lebih jelasnya lihat Gambar 29.



Gambar 29. Implementasi Tabel Ortu

D. Tahap Pengujian

Pada tahap ini aplikasi yang telah dikembangkan diberikan rangkaian pengujian kualitas perangkat lunak untuk mengetahui kualitas perangkat lunak yang dikembangkan. Dalam pengujian ini menggunakan beberapa instrumen pengujian perangkat lunak di antaranya : *Correctness, Maintainability, Integrity, Usability, Efficiency, Reliability, Interoperability* dan *Portability*.

1. Hasil pengujian Correctness

Pengujian faktor kualitas perangkat lunak *correctness* dapat dilakukan dengan menghitung *defect* per KLOC (cacat/ *error* pada setiap *Kilo Line of Code / KLOC*). Untuk mendapatkan jumlah KLOC dalam *project php* dapat dihitung menggunakan *Sourch Line of Code Counter*. Kemudian hasil KLOC di bandingkan dengan teori batas toleransi *defect/ error* per KLOC yang sudah ada.

a) Penghitungan *Line of Code/ LOC*

Penghitungan *Line of Code* dilakukan menggunakan *software Sourch Line of Code Counter*, dengan hasil seperti Gambar 30.

Source Line of Code Counter	
File Name	Lines
C:\xampp\htdocs\ci_sicmax\admin\error_log	1
C:\xampp\htdocs\ci_sicmax\admin\index.php	1
C:\xampp\htdocs\ci_sicmax\application\controllers\index.html	10
C:\xampp\htdocs\ci_sicmax\application\controllers\welcome.php	27
C:\xampp\htdocs\ci_sicmax\application\controllers\admin\admin.php	116
C:\xampp\htdocs\ci_sicmax\application\controllers\admin\data.php	354
C:\xampp\htdocs\ci_sicmax\application\controllers\admin\login.php	82
C:\xampp\htdocs\ci_sicmax\application\controllers\admin\report.php	48
C:\xampp\htdocs\ci_sicmax\application\controllers\admin\stat.php	147
C:\xampp\htdocs\ci_sicmax\application\controllers\user\coba.php	14
C:\xampp\htdocs\ci_sicmax\application\controllers\user\data.php	116
C:\xampp\htdocs\ci_sicmax\application\controllers\user\home.php	8
C:\xampp\htdocs\ci_sicmax\application\controllers\user\login.php	82
C:\xampp\htdocs\ci_sicmax\application\controllers\user\report.php	33
C:\xampp\htdocs\ci_sicmax\application\controllers\user\stat.php	153
C:\xampp\htdocs\ci_sicmax\application\controllers\user\user.php	294
C:\xampp\htdocs\ci_sicmax\application\helpers\index.html	10
C:\xampp\htdocs\ci_sicmax\application\helpers\login_helper.php	43
C:\xampp\htdocs\ci_sicmax\application\helpers\logout_helper.php	10
5773 LOC. 85 files. 29 directories.	
Restart Save	
Done!	

Gambar 30. Hasil Pengujian KLOC dengan SLOC

Dari hasil penghitungan LOC menghasilkan 5773 LOC, atau dibulatkan 5,7 KLOC.

b) Penghitungan *defect* per KLOC

Penghitungan *defect* per KLOC dilakukan menggunakan rumus, penghitungan dilakukan menggunakan *Ms.Excel*, dengan memasukan rumus sesuai dengan metode para ahli. Untuk keterangan lebih lanjut dapat dilihat pada Tabel 8.

Table 8. Hasil Penghitungan KLOC

METODE	KLOC	Perkiraan Error (McConnel)	HASIL
Watson - Felix Model	5.7	Untuk baris code	25.34250042
Bailey-Basili Model	5.7	2K - 6K batas error 0 – 40 <i>error</i> / <i>KLOC</i>	10.99715133
Boehm Simple Model	5.7		19.89841924
Doty Model(untuk KLOC>9)	5.7		32.7108945

Dari hasil penghitungan KLOC kemudian dilakukan perbandingan dengan nilai toleransi batas *error*, sesuai dengan standar *defect/ error* per KLOC. Dari hasil penghitungan KLOC sebesar 5.7 KLOC, maka masuk dalam range 2K - 16K dengan batas toleransi *error* 0-40 *defect/ error* per KLOC, melihat hasil pengujian KLOC dari ke 4 teori penghitungan nilai KLOC, hasil penghitungan tidak lebih dari 40 *defect/ error* per KLOC sebagai batas maksimal toleransi jumlah *defect/ error* per KLOC. Sehingga dapat disimpulkan kualitas perangkat lunak dari segi *correctness* sudah memenuhi.

2. Hasil Pengujian Usability

Pengujian faktor kualitas perangkat lunak aspek *usability* dihitung menggunakan metode hitung kuisioner SUS (*System Usability Scale*) sesuai dengan kuisioner yang digunakan. Untuk lebih dijelaskan pada Tabel 9.

Table 9. Penghitungan Kuisisioner *System Usability Scale* (SUS)

R	1	2	3	4	5	6	7	8	9	10	Total	2.5	Hasil
1	3	4	4	4	2	4	4	3	3	4	35	2.5	87.5
2	4	1	4	4	3	3	3	4	3	4	33	2.5	82.5
3	3	0	4	4	2	4	4	4	4	3	32	2.5	80
4	4	1	3	4	4	2	4	4	2	4	32	2.5	80
5	4	2	4	3	1	3	4	4	4	3	32	2.5	80
6	3	3	4	4	3	4	4	3	3	2	33	2.5	82.5
7	4	3	3	3	3	3	4	4	3	2	32	2.5	80
8	4	3	3	4	3	3	4	3	2	3	32	2.5	80
9	4	3	3	3	4	3	4	3	3	3	33	2.5	82.5
10	3	4	3	3	3	3	4	4	3	4	34	2.5	85
11	3	3	4	4	3	2	4	4	3	4	34	2.5	85
12	3	1	3	3	3	3	4	3	4	3	30	2.5	75
13	3	2	3	4	3	3	4	3	3	4	32	2.5	80
14	3	1	3	4	3	3	4	4	2	3	30	2.5	75
15	4	3	4	4	3	3	3	3	4	3	34	2.5	85
16	2	2	4	3	3	2	4	3	4	3	30	2.5	75
17	2	3	3	3	3	4	4	3	2	3	30	2.5	75
18	3	3	3	4	3	3	3	4	2	4	32	2.5	80
19	2	4	4	3	4	3	4	4	2	3	33	2.5	82.5
20	2	3	4	3	4	3	4	3	2	3	31	2.5	77.5
21	4	2	4	1	4	3	4	4	4	4	34	2.5	85
22	3	3	3	3	4	4	4	4	4	3	35	2.5	87.5
23	3	3	3	2	3	4	3	4	3	2	30	2.5	75
24	4	3	4	2	4	3	4	4	4	2	34	2.5	85
25	3	2	3	3	4	2	3	4	3	3	30	2.5	75
26	3	4	3	2	3	4	4	4	4	3	34	2.5	85
27	4	3	3	4	4	2	4	4	4	3	35	2.5	87.5
28	4	3	4	3	4	2	3	4	3	4	34	2.5	85
29	4	3	4	4	3	3	3	3	4	3	34	2.5	85
30	3	3	4	4	3	3	4	2	3	4	33	2.5	82.5
Hasil Seluruh Responden												2442.5	
Jumlah Responden												30	
Rata-Rata (Nilai SUS)												81.41667	

*Ket : R = Responden

Dari penghitungan Tabel 9 di atas didapatkan nilai total angket **2442.5** yang dibulatkan menjadi **2443**. Selanjutnya penghitungan diambil rata-rata dari jumlah total nilai angket :

$$N = \text{Total Nilai} / \text{Total Responden}$$

$$N = 2443 / 30 = 81.4$$

Dari penghitungan nilai di atas kemudian dibandingkan dengan standar nilai *System Usability Scale* (SUS).

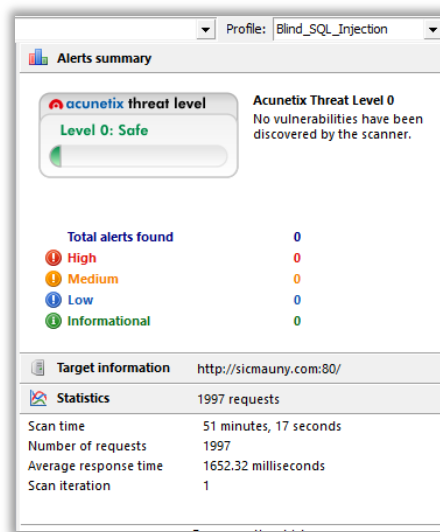
Table 10. Nilai Standar *System Usability Scale* (SUS)

SUS Score	Nilai (GRADE)
0-60	F
60-70	D
70-80	C
80-90	B
90-100	A

Berdasarkan standar *System Usability Scale* dapat dilihat bahwa aplikasi mendapat grade **B** dengan nilai **81.4**, Sehingga dapat dikatakan aplikasi lolos uji aspek *usability* dengan predikat **baik**.

3. Hasil pengujian Integrity

Pengujian aspek *integrity* dilakukan menggunakan menggunakan *Web Vulnerability Scanner*. Pengujian di fokuskan pada celah *website* dari sisi *SQL Injection*, *Blind_SQL Injection* dan *XSS* (Cross Site Scripting). Hasil pengujian dapat dilihat pada Gambar 31, berikut.



Gambar 31. Hasil Pengujian *Web Vulnerability Scanner*

Dari hasil pengujian di atas secara lebih detail pengujian dijelaskan pada Tabel

11.

Table 11. Hasil Pengujian Integrity (Web Vulnerability Scanner)

No	Security	Hasil
1	<i>SQL Injection</i>	Lolos Uji
2	XSS	Lolos Uji
3	<i>Blind_SQL Injection</i>	Lolos Uji

Pengujian selanjutnya dilakukan dengan metode *black-box testing* sebagai pendukung dari pengujian dari *integrity*, hal ini dilakukan untuk memenuhi syarat uji *valid* dan *reliable*. Pengujian *black-box testing* dilakukan dengan angket yang diuji oleh pengembang sendiri dengan hasil seperti pada Tabel 12.

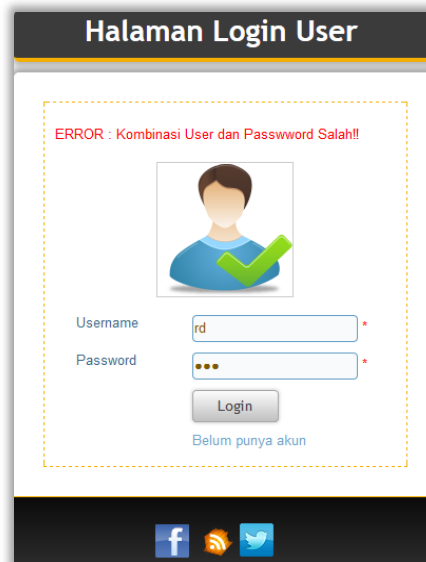
Table 12. Hasil Angket Pendukung Integrity Testing

No	Kuisisioner	Hasil
1	Apakah terdapat pembagian <i>user</i> dan hak akses dalam aplikasi ini ?	Ya
2	“ <i>super-admin</i> ” dapat menambah “ <i>admin</i> ” dan “ <i>user</i> ” baru ?	Ya
3	“ <i>super-admin</i> ” dapat mengakses halaman “ <i>admin</i> ” dan “ <i>user</i> ” ?	Ya
4	“ <i>super-admin</i> ” dapat melakukan perubahan data pada “ <i>admin</i> ” dan “ <i>user</i> ” ?	Ya
5	“ <i>super-admin</i> ” dapat melakukan pencarian data <i>user</i> (biasa) ?	Ya
6	“ <i>super-admin</i> ” dapat melakukan perubahan data “ <i>admin</i> ” dan “ <i>user</i> ” ?	Ya
7	“ <i>admin</i> ” dapat mengakses halaman <i>admin</i> ?	Ya
8	“ <i>admin</i> ” dapat melakukan pencarian data <i>user</i> (biasa)?	Ya
9	“ <i>admin</i> ” dapat melakukan perubahan data <i>user</i> (biasa)?	Ya
10	“ <i>user</i> “(biasa) tidak dapat mengakses halaman “ <i>admin</i> ”.	Ya
11	“ <i>user</i> ” (biasa) tidak dapat melakukan perubahan data “ <i>admin</i> ”	Ya
12	“ <i>user</i> ” (biasa) dapat mengubah data pribadi.	Ya
13	“ <i>user</i> ” (biasa) tidak dapat melakukan perubahan data pada <i>user</i> (biasa) lain.	Ya
14	“ <i>user</i> ” (biasa) dapat melakukan pencarian (<i>tracking</i>) dan melihat informasi <i>user</i> (biasa) lain.	Ya
15	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat mengakses halaman <i>web</i> kecuali halaman pendaftaran.	Ya
16	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat melihat data.	Ya
17	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat melakukan pencarian data.	Ya
18	“ <i>user</i> ” (tidak <i>login</i>) tidak dapat melakukan perubahan data	Ya

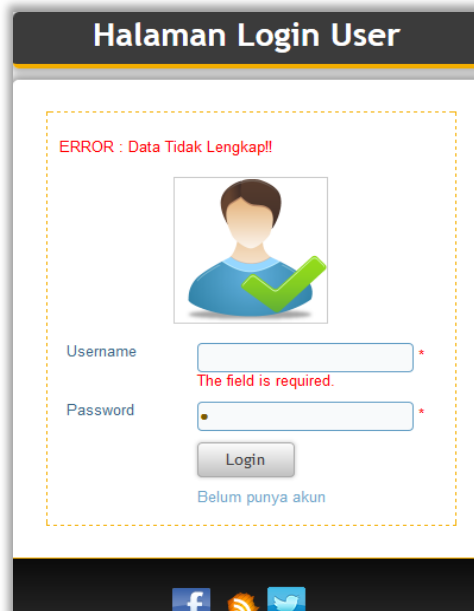
Pengujian yang dilakukan terbukti bahwa *software* lolos uji analisis dari segi *SQL Injection* dan *XSS* dan *Blind_SQL Injection* maupun uji tester *user*.

4. Hasil pengujian Maintainability

Pengujian pada aspek *maintainability* dilakukan dengan menggunakan ukuran–ukuran (metrik), yang selanjutnya pengujian dilakukan secara operasional dengan hasil pengujian seperti pada Gambar 32 dan Gambar 33.

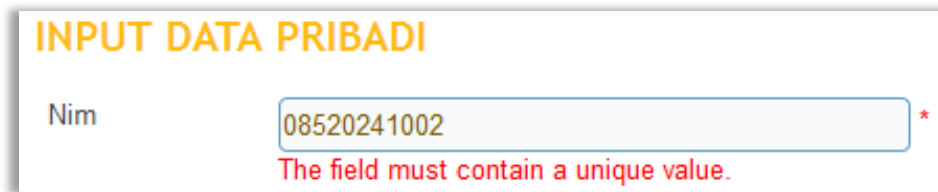


Gambar 32. Peringatan Gagal Login (*username* dan *password* salah)



Gambar 33. Peringatan Gagal Login

Pada form login terdapat 2 jenis notifikasi kesalahan. Untuk kesalahan kecocokan *username* dan *password* notifikasi : “ ERROR : Kombinasi *username* dan *password* Salah”, sedangkan untuk notifikasi pengisian tidak lengkap notifikasi : “ ERROR : Data Tidak Lengkap !!”, dan terdapat notifikasi tambahan pada form yang belum diisi “*This field is required*”.



INPUT DATA PRIBADI

Nim *

The field must contain a unique value.

Gambar 34. Peringatan saat Registrasi Mandiri

Pada *form input data* dan *edit data* terdapat *form validasi* untuk memvalidasi keunikan data, contoh diatas merupakan notifikasi kesalahan input data karena *nim* yang dimasukkan sudah ada dalam *database*, hal ini digunakan untuk menghindari adanya duplikat data pada *database* yang dapat mengakibatkan kerusakan/ *error system*.

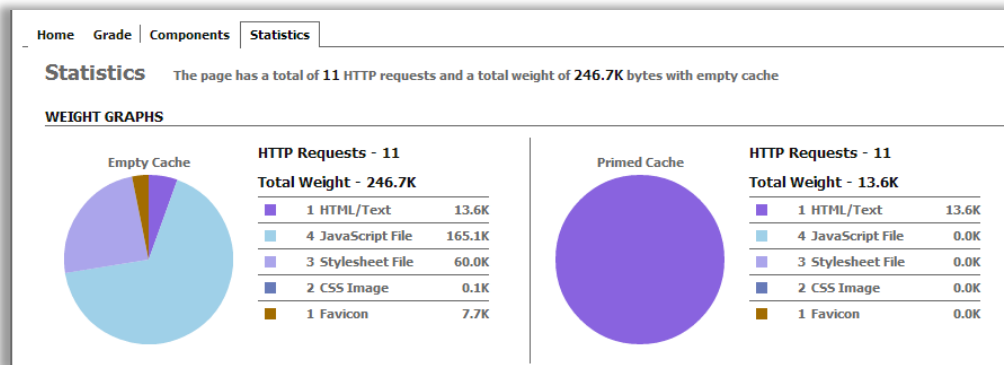
Dari hasil *load tester* di atas kemudian analisis pengujian diimplementasikan dalam *metrics* yang digunakan untuk menguji aspek *maintainability*.

Table 13. Hasil Snalisis Aspek Maintainability

Aspek	Aspek yang dinilai	Hasil yang diperoleh
<i>Instrumentation</i>	Terdapat peringatan pada sistem untuk mengidentifikasi kesalahan	Hasil pengujian yang telah dilakukan peneliti menunjukkan bahwa ketika ada kesalahan yang dilakukan oleh user, maka sistem akan mengeluarkan peringatan untuk mengidentifikasi kesalahan. Contoh, ketika user menginputkan data baru, dan masih ada data yang kosong maka akan muncul peringatan data apa yang masih kosong.
<i>Consistency</i>	Penggunaan satu bentuk rancangan pada seluruh rancangan sistem	Hasil pengujian menunjukkan bahwa bentuk rancangan sistem mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem, dimana tampilan halaman web dari satu halaman ke halaman lainnya memiliki kemiripan, bentuk yang serupa, dan konsisten.
<i>Simplicity</i>	Kemudahan dalam pengelolaan, perbaikan, dan pengembangan sistem	Hasil pengujian menunjukan bahwa sistem mudah untuk dikembangkan, karena menggunakan <i>Framework PHP Codeignitier</i> yang berbasis <i>Model-View-ControlerI</i> (MVC). Ketika ditemukan kegagalan fungsi dapat ditelusuri pada bagian modul/ <i>controler</i> program sehingga memudahkan perbaikan sistem. Misalkan terdapat kesalahan pada bagian <i>input</i> data, maka pengembang hanya perlu mencari dan memperbaiki pada bagian <i>controller input</i> data. Untuk pengembangan sistem dapat dilakukan dengan menambahkan <i>controller</i> atau fungsi pada <i>controller</i> tanpa merubah fungsi yang lain(<i>modular</i>)

5. Hasil pengujian Efficiency

Hasil pengujian halaman data dengan mode *admin* dilakukan dengan menggunakan *YSlow*, dengan HTTP request 11 buah dan besarnya dokumen 13.6kb setelah dilakukan kompresi pada *javascript* dan *css* .



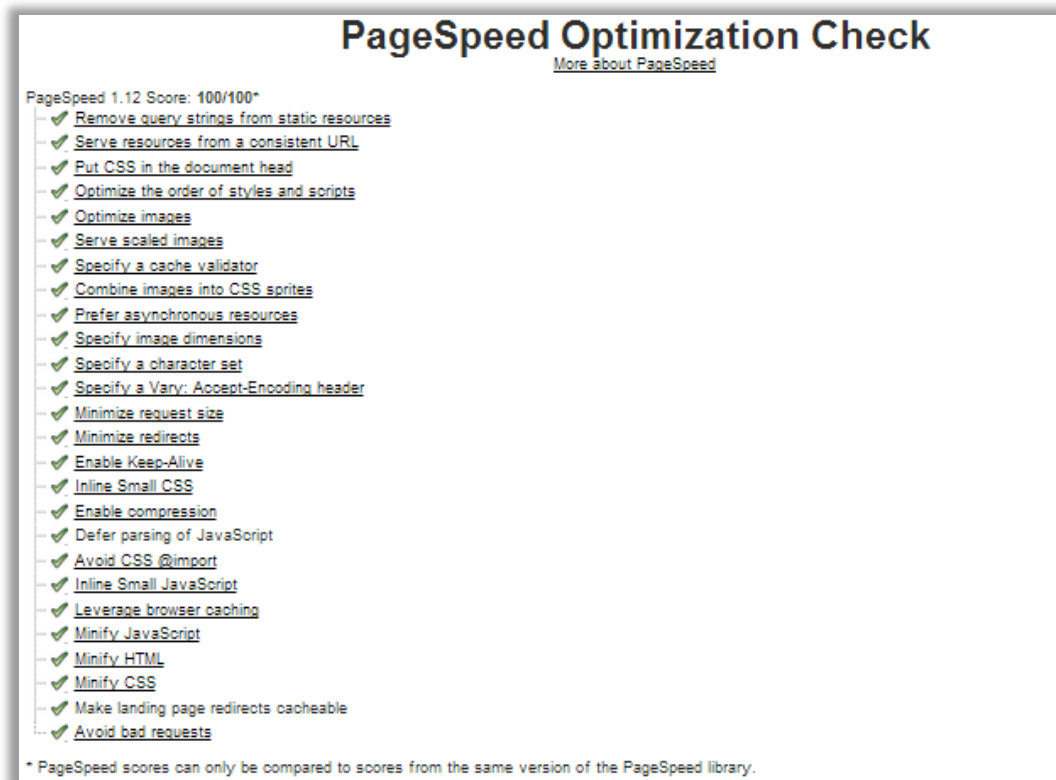
Gambar 35. Hasil Uji Halaman data (admin mode)

Dari pengujian performance juga memberikan hasil *grade* pengujian. Untuk lebih jelasnya dapat dilihat pada Gambar 36.



Gambar 36. Hasil Grade YSlow

Selanjutnya pengujian dilakukan menggunakan *webpagetest* sebuah web testing yang menguji *performance* web yang dikembangkan oleh google. Hasil pengujian dapat dilihat pada Gambar 37.



Gambar 37. Pengujian Performance Web dengan Webpagetest

Dari gambar diatas terlihat bahwa website mendapatkan nilai 100/100 yang berarti nilai penuh untuk *performance web*. Dari pengujian performa menggunakan *YSlow* dengan *grade A* dan *Webpagetest* dengan nilai 100/100 menunjukkan aplikasi sudah memenuhi dari segi *efficiency*.

6. Hasil pengujian Reliability

Pengujian reliability dilakukan dengan menggunakan tools *software* [Xenu Link Sleuth 1.3.8](#). untuk pengujian *Correct link processing*, dan pengujian *User input validation and recovery*, dan *Error recovery* dilakukan *load tester* pada *browser*.

a) Pengujian Correct link processing

Hasil pengujian *correct link processing* menggunakan *Xenu Link Sleuth 1.3.8* dapat dilihat pada Gambar 38.

Total	43 URLs	100.00 %	46416 9 Bytes	(45 3 KB)	100.00 %	
-------	------------	-------------	---------------------	-----------------	-------------	--

All pages, by result type:

ok	43 URLs	100.00%
Total	43 URLs	100.00%

[Return to Top](#)

This report has been created with [Xenu Link Sleuth 1.3.8](#)

Gambar 38. Hasil pengujian Xenu Link Sleuth 1.3.8.

b) Pengujian User input validation and recovery, dan Error recovery

Pengujian ini dilakukan secara *load tester* pada *browser* sebagai pembuktian *validation input* dan *recovery*. Dengan hasil pada Gambar 39 dan Gambar 40.

Halaman Login User

Username *
The field is required.

Password *
The field is required.

Login

[Belum punya akun](#)

UKM Band SICMA Universitas Negeri Yogyakarta

Gambar 39. Hasil Uji Validasi Login User

Validasi data pada *form login* jika *form username* dan *password* tidak diisi.

Halaman Login User

Username *
The field may only contain alphabetical characters.

Password *

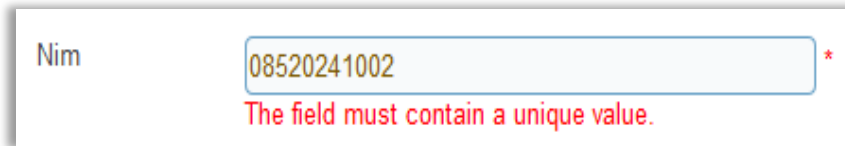
Login

[Belum punya akun](#)

Gambar 40. Hasil Pengujian Login User (2)

Validasi pada saat *login* dengan membatasi karakter *input* dengan hanya menerima karakter huruf (*alphabetical*) sehingga terhindar dari karakter *input* data

yang dapat mengakibatkan *error database* dan *system* misalnya penggunaan karakter petik ('). Seperti pada Gambar 41.

A screenshot of a web form with a label 'Nim' and a text input field. The input field contains the value '08520241002'. To the right of the input field is a red asterisk. Below the input field, a red error message reads 'The field must contain a unique value.'

Gambar 41. Hasil Pengujian Form Nim (Input Data)

Pada pengisian *form nim*, terdapat *validasi duplikat* data, dan jika memasukkan data *nim* yang telah terdaftar pada *database* maka akan muncul *warning* berupa *notifikasi* bahwa *nim* harus *unique value* sehingga *user* baru tidak dapat memasukkan data *nim* yang telah terdaftar dalam *database* dengan demikian dapat mencegah terjadinya duplikasi data pada *database* yang dapat menyebabkan *error system*. Selain itu dengan adanya *recovery data input error* masih ditampilkan kembali guna mencegah *user* memasukkan data salah yang sama.

7. Hasil pengujian Interoperability

Pengujian *interoperability* dilakukan menggunakan *metric*. *Metric* yang digunakan dalam pengujian aspek *interoperability* menggunakan cohesion metrik dengan indikator pengujian untuk *interoperability* adalah LCC dan *coupling metric* dengan matric pengujian NOC, CBO dan RFC.

a) Pengukuran LCC

Pengukuran LCC adalah pengukuran hubungan antara *method* atau *function* dalam *class* dengan persamaan hitung $LCC = (NDC + NID) / NP$. NDC adalah

jumlah koneksi langsung, NID jumlah koneksi tidak langsung, dan NP adalah jumlah koneksi maksimal dengan rumus $N*(N-1)/2$ dengan N adalah *method* atau *function*. Untuk lebih jelasnya dapat dilihat pada Gambar 42.

```
function view() {
    $query = $this->m_admin->view();
    $query = $query->result_array();
    $data['query'] = $query;

    $this->load->view('css/admin/header');
    $this->load->view('css/admin/menu');
    $this->load->view('admin/v_viewAdmin', $data);
    $this->load->view('css/admin/footer');
}

function delete($id) {
    if (!is_admin_super())
        redirect('admin/admin/view');
    $query = $this->m_admin->delete($id);
    redirect('admin/admin/view');
```

Gambar 42. Gambar Pengukuran LCC

Pada gambar 40 diatas *function delete* mengakses langsung *function view* pada class admin. Dengan demikian setidaknya terdapat 1 hubungan langsung NOC pada *class admin*. Untuk hasil penghitungan LCC dapat dilihat pada Tabel 14.

Table 14. Hasil Penghitungan LCC

Class	N	NDC	NID	NP	LCC
CI_Controller/admin	28	6	0	378	0.016
CI_Controller/user	19	3	0	171	0.017
CI_Model/admin	39	0	0	741	0
CI_Model/user	40	0	0	780	0
Helper	2	0	0	1	0
Rata-rata					0.0067

b) Pengukuran CBO

Pengukuran CBO dilakukan dengan menghitung jumlah kelas yang berpasangan, dimana kelas yang dihitung tidak mempunyai hubungan *inheritence* atau pewarisan. Pengukuran pasangan kelas dihitung jika pasangan kelas menggunakan *object* atau *attribut* yang sama. Seperti pada contoh Gambar 43.

```
function update() {  
    if (!is_login())  
        redirect('user/login');  
  
    $username_ = $this->session->userdata('username');  
    $query = $this->m_user->cek_id($username_);  
    $query = $query->row_array();  
}
```

Gambar 43. Pengukuran CBO

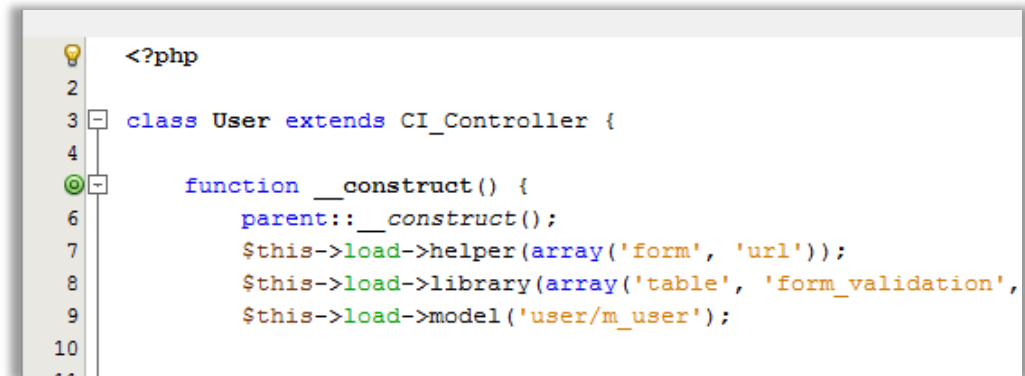
Dari Gambar 43 di atas dapat dilihat bahwa *function view_profil* pada kelas *user* mengakses kelas *m_user* pada *function view_detail* dengan *\$username_* sebagai *attribute* atau *variable* yang menghubungkan. Dengan demikian dapat dikatakan terdapat pasangan kelas pada kelas *user* dan *m_user*. Untuk lebih jelasnya dapat dilihat pada Tabel 15.

Table 15. Hasil Pengukuran CBO

Class	Method	CBO
CI_Controller/admin	28	5
CI_Controller/user	19	4
CI_Model/ admin	39	5
CI_Model/user	40	4
Helper	2	1
Rata-rata		3.8

c) Pengukuran NOC

Pengukuran NOC adalah menghitung jumlah *subclass* dari kelas *parent*. Salah satu ciri kelas anak atau *child* menggunakan *extends* didepan nama *class*, seperti pada contoh Gambar 44.



```
<?php
2
3 class User extends CI_Controller {
4
5     function __construct() {
6         parent::__construct();
7         $this->load->helper(array('form', 'url'));
8         $this->load->library(array('table', 'form_validation',
9         $this->load->model('user/m_user');
10
11
```

Gambar 44. Pengukuran NOC

Pada Gambar 44 di atas terlihat bahwa kelas *user* merupakan anak kelas dari kelas *CI_Controller*. Hal ini ditunjukkan dengan adanya *extends* di depan *class user*. Untuk hasil selengkapnya dari pengukuran NOC dapat dilihat pada Tabel 16.

Table 16. Hasil Pengukuran NOC

Class	Hasil
CI_Controller/admin	5
CI_Controller/user	4
CI_Model/admin	5
CI_Model/user	4
Helper	0
Rata- rata	1.6

d) Pengukuran RFC

Pengukuran pada RFC dilakukan pada setiap fungsi pada kelas yang memanggil fungsi pada kelas lain. Untuk lebih jelasnya dapat dilihat pada Gambar 45.

```
function view_profil() {  
    if (!is_login())  
        redirect('user/login');  
    $query = $this->m_user->view_detail();  
    $query = $query->row_array();  
    $data['query'] = $query;  
}
```

Gambar 45. Gambar Fungsi RFC

Pada Gambar 45 di atas *function view_profil* memanggil *function view_detail* dari *class m_user*, panggilan ini dihitung sebagai 1 panggilan RFC. Untuk hasil keseluruhan dapat dilihat pada Tabel 17.

Table 17. Hasil Pengukuran RFC

Class	Method	RFC
CI_Controller/admin	28	47
CI_Controller/user	19	36
CI_Model/admin	39	0
CI_Model/user	40	0
Helper	2	4
Rata-rata		17.4

Dari hasil pengujian *LLC*, *CBO*, *NOC*, dan *RFC* di atas kemudian dibandingkan dengan nilai yang direkomendasikan dalam *score metric* (Chandra & Linda, 2010), seperti di jelaskan pada Tabel 18.

Table 18. Hasil Pengujian Metrik Interoperability





Metric	Batas Nilai	Hasil
LCC	0-1	0.0067
CBO	0-8	3.8
NOC	0-6	4
RFC	0-35	17.47

Dari hasil perbandingan pengukuran *metric* LCC, CBO, NOC, dan RFC di atas menunjukkan bahwa hasil pengukuran semua *metric* sesuai dengan nilai yang direkomendasikan. Dengan demikian dapat dikatakan pengujian pada aspek *interoperability* sudah memenuhi.

8. Hasil Pengujian *Portability*

Pengujian pada aspek *portability* dilakukan dengan melakukan *running* sistem pada *browser* yang berbasis *dekstop* maupun *browser* berbasis *mobile*. Untuk lebih jelasnya dapat dilihat pada Tabel 19.

Table 19. Hasil Pengujian Portability

No.	Browser	Tampilan	Error
1.	Mozilla Firefox		Tidak ditemukan <i>error</i> .
2.	Internet Explorer		Tidak ditemukan <i>error</i> .
3.	Google Chrome		Tidak ditemukan <i>error</i> .
4.	Opera		Tidak ditemukan <i>error</i> .

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa:

1. hasil dari pengembangan perangkat lunak adalah berupa Sistem Informasi Pengelolaan Data Anggota UKM Sicma UNY berbasis *Codeigniter PHP Framework*, perangkat lunak tersebut telah memenuhi kebutuhan pihak UKM Sicma UNY;
2. dapat diketahui kualitas perangkat lunak berbasis aplikasi *web* yang dikembangkan berdasarkan standar kualitas perangkat lunak (*Software Quality*) , yaitu:
 - a) Kualitas perangkat lunak yang dikembangkan dari sisi *Correctness* sudah memenuhi, dengan tidak adanya *error code* dalam program;
 - b) Kualitas perangkat lunak yang dikembangkan dari sisi *Usability* sudah memenuhi dengan hasil pengujian *software* mendapat grade B (baik) dari standar *grade A-F* dalam pengujian menggunakan *System Usability Scale*;
 - c) Kualitas perangkat lunak dari *integrity* yang memfokuskan pada tingkat keamanan sudah memenuhi, dengan hasil tingkat keamanan *software* lolos uji dari serangan *SQL Injection*, *XSS*, *Blind_SQL Injection*;

- d) Kualitas perangkat lunak dari sisi *maintainability* lolos uji sesuai pembuktian hasil *software* dan penjelasan dari *metric* penilaian *maintainability*;
- e) Kualitas perangkat lunak dari sisi *efficiency* yang diuji dengan *Yslow* dan *Google Page View* sudah memenuhi dengan mendapatkan *grade A* dan nilai 100/100;
- f) Kualitas perangkat lunak dari sisi *reliability* diuji dengan *Xenu Link Slueth* dengan hasil 100% link bekerja dan pembuktian secara *load tester* pada *User input validation and recovery*, dan *Error recovery* terdapat validasi yang dibutuhkan;
- g) Kualitas perangkat lunak dari segi *interoperability* dengan pengujian *cohesion metric*(LCC) dan *coupling metric* (NOC, CBO dan RFC) sudah memenuhi standar penilaian.
- h) Kualitas perangkat lunak dari segi *portabilty* sudah memenuhi standar dan lolos uji dengan dilakukannya *running test* sistem pada *browser* yang berbasis *dekstop error*;

B. Saran

Mengingat berbagai keterbatasan yang dimiliki penulis baik dari sisi pemikiran dan waktu, maka penulis menyarankan untuk pengembangan penelitian di masa yang akan datang sebagai berikut :

- 1) Fitur pengelolaan data anggota UKM yang lebih beragam.
- 2) Teknik pengujian perangkat lunak yang lebih beragam dan mengungkap kualitas perangkat lunak.

DAFTAR PUSTAKA

- A.S, R., & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Ali, M. (2010). *Metodologi dan Aplikasi Riset Pendidikan*. Bandung: Pustaka Cendekia Utama.
- Bevan, N. (2006). *International Standards for HCI*. London: Idea Group.
- Brooke, J. (1996). *SUS - A quick and dirty usability scale*. United Kingdom: Redhatch Consulting Ltd.,.
- Chandra, D. E., & Linda, P. E. (2010). Class Break Point Determination Using CK Metrics. *Global Journal of Computer Science and Technology*, 2.
- Darmastuti, D. (2013). *Implementasi Metode Simple Additive Weighting (SAW) dalam Sistem Informasi Lowongan Kerja Berbasis Web untuk Rekomendasi Pencari Kerja Terbaik*, 2.
- Earl A. Lund, P. E. (1993). *Technical Circular T-6/93* . Branch & Ministry.
- F.Sumway, M. (1997). *Mesuring Class Cohesien in Java*. Colorado: Computer Science Departement Colorado State University.
- Folmer, I. E., & Verhoosel, D. i. (t.thn.). *State of the Art on Semantic IS Standardization, Interoperability & Quality*. (J. S.-K.-P. Services, Penyunt.)
- Ginting, S. L. (2010). *Faktor-faktor Kualitas Perangkat Lunak*. Bandung: Jurusan Teknik Komputer Fakultas Teknik dan Ilmu Komputer (UNIKOM).
- google. (2008). *about*. Diambil kembali dari webpagetest: <http://www.webpagetest.org/about>
- Hanggara, Y. (2012). 1. Analisis Sistem Informasi Pengelolaan Data Alumni Sekolah Berbasis Codeigniter PHP Framework.
- Land, R. (t.thn.). *Measurements of Software Maintainability*. Vasteras, Sweden : Mälardalen University Department of Computer Engineering .
- Legner & Lebreton, Rukanova. (2007). *State of the Art on Semantic IS Standardization*,.
- Luis Olsina, Gustavo Rossi. (2001). A Quantitative Method for Quality Evaluation of Web Sites and.

- Marcelduran. (2013). *YSlow faq*. (Yahoo, Produser, & Yahoo) Dipetik juni 2013, dari YSlow: <http://yslow.org/faq/>
- McCall, R. &. (1977). *Factors in Software* . US Rome Air Development Center Reports.
- McConnell, S. (2004). *Code Complete : A Practical Handbook of Software Construction*. Washington: Microsoft Press.
- Mohammed, N., & Munassar, A. (2010). A Comparison Between Five Models Of Software Engineering. *IJCSI International Journal of Computer Science Issues*, Vol. 7, 95.
- Nielsen's, J. (t.thn.). Jakob Nielsen's Alertbox . *Usability 101: Introduction to Usability* , 1.
- Olsina, Roger S. Pressman (5th ed.). (2001). *Query Recuirements tree* (Vol. 888). Avenue of Americas, New York: McGraw-Hill.
- Paasonen, T. (2011). *Methods for Improving the Maintainability of Application Software*. Aalto University.
- Pressman, R. S. (2010). *Software Engineering A Practitioner's Approach-7ed*. New York: Raghothaman Srinivasan.
- Rakshith V, R. S. (2011). International Journal of Systems , Algorithms & Applications (IJSAA). *Implementation of Conceptual Cohesion of Classes to Predict Faults in Object*, 2.
- Rodriguez, D., & Harrison, R. (2001). *An Overview of Object-Oriented Design Metrics*. RUCS/2001/TR/A.
- Roger S.Pressman, P. (2001). *Software Enginering - A Practitioner Approach*. New York: Thomas Casson.
- Rosenberg, D. L. (t.thn.). *Applying and Interpreting Object Oriented Metrics*. Greenbelt: Linda H. Rosenberg, Ph.D.
- S.Pressman, R. (2001). *Software Enginering - A Practitioner Approach* (5 ed.). New York, America: Thomas Casson.
- Saradhi, M. V. (2010). Journal of Engineering Science and Technology. *A Quality Indicator for Software International*, 3.
- Sutabri, T. (2012). *Analisis Sistem Informasi*. Yogyakarta: Andi Offset.
- Tham, C. (2007). *about*. Diambil kembali dari Code Central: <http://thecodecentral.com/about>

Wiswakarma, K. (2010). *9 Langkah Menjadi Master Framework codeigniter*. (L. Hakim, Penyunt.) Bekasi, Jawa Barat: Penerbit Lokomedia.

LAMPIRAN

A. System Usability Scale (SUS)

**ANALISIS SISTEM INFORMASI PENGOLAHAN DATA ANGGOTA UKM
BERBASIS CODEIGNITER PHP FRAMEWORK DAN MYSQL**

Nama : Chandra Mussana

Fakultas / Prodi : FT / Ind. Teknik Sipil

Analisis : Usability Testing menggunakan SUS (System Usability Scale)

Uji lapangan : 30 responden Mahasiswa

Peneliti dan pengembang : Faris Ridha Primastomo

Tanggal : 29 Oktober 2013

Petunjuk :

- Lembar evaluasi diisi oleh responden (Mahasiswa UKM)
- Lembar evaluasi digunakan untuk mendapatkan hasil evaluasi kualitas perangkat lunak system pengolahan data anggota ukm berbasis php dan mysql yang dikembangkan dari segi *usability*.
- Pilihlah jawaban yang sesuai dengan pilihan anda dengan memberikan tanda (V), kuisisioner berisi 10 pertanyaan dengan 5 range titik skala dari "Sangat tidak setuju – Sangat setuju".

Keterangan :

1 = Sangat Tidak Setuju (STS)
2 = Tidak Setuju (TS)
3 = Ragu-ragu (RG)
4 = Setuju (ST)
5 = Sangat Setuju (SS)

d. Mahasiswa responden dimohon memberikan kometar dan saran pada tempat yang telah ditentukan

Lampiran 1. Contoh Kuisisioner System Usability Scale (1)

Lembar pengisian :

④

NO	Kuisiner	STS	TS	RG	ST	SS
1	Saya berpikir untuk sering menggunakan website ini					✓
2	Menurut saya website ini tidak terlalu kompleks				✓	
3	Menurut saya website ini mudah digunakan				✓	
4	Saya membutuhkan trainer untuk menggunakan website ini	✓				
5	Saya menemukan berbagai fungsi website ini terintegrasi dengan baik					✓
6	Saya pikir ada inkonsistensi terlalu banyak di website ini			✓		
7	Menurut saya kebanyakan orang akan dengan mudah mempelajari penggunaan situs ini					✓
8	Menurut saya website ini sangat rumit digunakan	✓				
9	Saya merasa sangat percaya pada website ini			✓		
10	Saya perlu belajar banyak tentang situs ini sebelum saya bisa menggunakannya dengan efektif	✓				

5
1
3
5
5
2
5
5
2
5

$$\begin{array}{r}
 1 \times 1 = 1 \\
 2 \times 2 = 4 \\
 3 \times 1 = 3 \\
 4 \times 6 = 24 \\
 \hline
 32 \\
 205 \\
 \hline
 160 \\
 69 \\
 \hline
 890
 \end{array}$$

80

Lampiran 2. Contoh Kuisiner System Usability Scale (2)

B. Perancangan Tampilan

A diagram of a login form layout. It consists of a large outer rectangle containing a smaller rectangle at the top labeled "Header". Below the header, there are two horizontal input fields stacked vertically. To the right of the bottom input field is a button labeled "Login". Below the input fields and button is the text "Link Pendfatraran Mandiri".

Lampiran 3. Perancangan Form Login

A diagram of a content form layout. It features a header bar at the top with "Header" on the left and "Menu" on the right. Below the header, there is a sidebar on the left labeled "Menu" and a large main area labeled "Form Pengisian Data". At the bottom, there is a footer bar labeled "Footer".

Lampiran 4. Perancangan Form Isi Konten

C. Implementasi Tampilan

Halaman Login User

Username

Password

Login

[Belum punya akun](#)

UKM Band SICMA Universitas Negeri Yogyakarta

Lampiran 5. Gambar Tampilan Login User

PENDATAAN ANGGOTA UKM BAND SICMA
UNIVERSITAS NEGERI YOGYAKARTA

ACCOUNT

MENU

Data

Statistik

Contact

Data Anggota "gondil"

DATA PROFIL

Nim : 08520241002

Nama : ridho faris primastomo

Nama panggilan : gondil

Tempat lahir : sleman

Tgl. lahir : 1989-12-12

Jenis kelamin : Laki-laki

Agama : Islam

Hp : 08562894489

Email : sleeping.anthem@gmail.com

Fakultas : TEKNIK

Prodi : 1

Alamat yogya : sleman

Alamat asal : cibukan sumberadi mlati sleman

Tahun masuk : 2008

DATA ORGANISASI ANDA

Player : Drummer

EO : Event Organizer

DATA ORANGTUA ANDA

Nama Orangtua : sujarno sdd

Hp Orangtua : 123435455343

Alamat Orangtua : cibukan sumberadi

Untuk Mengubah data diri silahkan klik link [Edit Profil](#)

Copyright_jun © 2011-2012

Lampiran 6. Gambar Tampilan Profil

<div>  <div> PENDATAAN ANGGOTA UKM BAND SICMA <small>UNIVERSITAS NEGERI YOGYAKARTA</small> </div> </div>						
ACCOUNT						
<div> <div> <div>Menu</div> <div> Data Statistik Contact </div> </div> <div> <div>Data Anggota</div> <div> <input type="text"/> <div>Search</div> </div> </div> </div>						
No	Nim	Nama	Nama Panggilan	CP	Email	Aksi
1	0620041002	ridho faris primaasmo	pondi	08262664466	aleevinganham@gmail.com	Detail
2	0677369466266	chlanul lyna	lyna	85643766022	lyna@gmail.com	Detail
3	6	bovo	bovo alah	08746466026	bovo@gmail.com	Detail
4	067667667	arifgians	ayuf	8736266260	ar_l@gmail.com	Detail
5	07516044001	Taufiq Fada Irdana	Taufiq	06160417066	taufiqfada@gmail.com	Detail
6	06766440260	slcma	slcma	0667664432	slcma@slcma.com	Detail
7	06667662736	slcmajunior	slcmajunior	0667667264	slcmajunior@slcmajunior.com	Detail
8	0620041004	Yoga hanggara	yoga	08766672646	yoga@gmail.com	Detail
9	0620041002	Muhamad Rizki Aji	Rizki	08266667666	rtgira@gmail.com	Detail
10	0620041001	adityanuar	ponari	06674666267	adit@gmail.com	Detail
11	06266194004	PUNDU	PUNDU	06266440027		Detail
12	10007264011	Dandi Hlmi	icang	06746210009	danxvengeance@yahoo.com	Detail
13	10006264020	astrah hakim nur lalla	lallah	06736626626	astrah105@yahoo.com	Detail
14	10206264011	RINTO RUMURJO OTONIK		06642661212	rinto_crispy@gmail.com	Detail
15	11206264036	ghnanjar aldik	anjar	06662066661	ghnanjar_aldik@yahoo.com	Detail
16	06201264066	Bagus Ady Kumawan	Bag	+62 827 2600 60	bagus_slcma@yahoo.co.id	Detail
17	06412144016	ryan galih ajuhade	ryan	062643766766	ryan_galih1@yahoo.com	Detail
18	11412141040	Muhamad Saleh A.	Saleh	06667212622	msalehas@gmail.com	Detail
19	12501264016	Julian Rinjani Pura	Julian	06266641464	julianrinjani@rockemail.com	Detail
20	06201261026	Yulan Angga Pradi	Yul	06272660027	yu_du@yahoo.com	Detail
1 2 3 4 Next						
Copyright © 2011-2012						

Lampiran 7. Gambar Tampilan *view_data_all*

D. Perancangan Database

Tables_in_heregistrasi
admin
agama
coba
data
eo
fakultas
konten
organisasi
ortu
player
prodi
user

Lampiran 8. Gambar Daftar Tabel Database

```
mysql> desc admin
-> ;
```

Field	Type	Null	Key	Default	Extra
idadmin	int(11)	NO	PRI	NULL	auto_increment
username_	varchar(45)	NO	PRI	NULL	
password_	varchar(45)	YES		NULL	
keterangan_	enum('super','biasa')	YES		NULL	

```
4 rows in set (0.00 sec)
```

Lampiran 9. Gambar Tabel Database "Admin"

```
mysql> desc user;
```

Field	Type	Null	Key	Default	Extra
username_	varchar(45)	NO	PRI	NULL	
pass_	varchar(45)	NO		NULL	
email_	varchar(45)	YES		NULL	

```
3 rows in set (0.00 sec)
```

Lampiran 10. Daftar Tabel Database "User"

```
mysql> desc data;
```

Field	Type	Null	Key	Default	Extra
no_increment	int(11)	NO	PRI	NULL	auto_increment
username_	varchar(45)	NO	PRI	NULL	
nim_	varbinary(15)	YES		NULL	
nama_	varchar(45)	YES		NULL	
nama_panggilan_	varchar(45)	YES		NULL	
tmpt_lahir_	varchar(45)	YES		NULL	
tgl_lahir_	date	YES		NULL	
jenis_kelamin_	enum('Laki-laki','Perempuan')	YES		NULL	
agama_	int(11)	YES	MUL	NULL	
hp_	varbinary(15)	YES		NULL	
email_	varchar(45)	YES		NULL	
idFakultas	int(3)	YES	MUL	NULL	
idProdi	varchar(100)	YES		NULL	
alamat_jogja_	varchar(100)	YES		NULL	
alamat_asl_	varchar(100)	YES		NULL	
tahun_masuk_	char(11)	NO		NULL	

```
16 rows in set (0.01 sec)
```

Lampiran 11. Daftar Tabel Database "Data"

```
mysql> desc fakultas;
```

Field	Type	Null	Key	Default	Extra
idFakultas	int(3)	NO	PRI	NULL	
fakultas	varchar(50)	YES		NULL	

```
2 rows in set (0.00 sec)
```

Lampiran 12. Daftar Tabel Database "Fakultas"

```
mysql> desc prodi
-> ;
```

Field	Type	Null	Key	Default	Extra
idProdi	int(5)	NO	PRI	NULL	auto_increment
prodi	varchar(50)	YES		NULL	
idFakultas	int(3)	YES		NULL	

```
3 rows in set (0.00 sec)
```

Lampiran 13. Daftar Database Tabel "Prodi"

```
mysql> desc organisasi;
```

Field	Type	Null	Key	Default	Extra
username_	varchar(45)	NO	PRI	NULL	
id_player	int(11)	YES	MUL	NULL	
id_eo	int(11)	YES	MUL	NULL	

```
3 rows in set (0.00 sec)
```

Lampiran 14. Daftar Tabel Database "Organisasi"

```
mysql> desc player;
```

Field	Type	Null	Key	Default	Extra
id_player	int(11)	NO	PRI	NULL	auto_increment
player_	varchar(45)	YES		NULL	

```
2 rows in set (0.00 sec)
```

Lampiran 15. Daftar Tabel Database "Player"


```
mysql> desc eo
-> ;
```

Field	Type	Null	Key	Default	Extra
id_eo	int(11)	NO	PRI	NULL	
eo_	varchar(45)	YES		NULL	

```
2 rows in set (0.00 sec)
```

Lampiran 16. Daftar Tabel Database "Eo"

